UNIVERSITY OF ZAGREB
Faculty of Mechanical Engineering and Naval Architecture

MASTER'S THESIS

Borna Šojat

Zagreb, 2017

UNIVERSITY OF ZAGREB
Faculty of Mechanical Engineering and Naval Architecture

DEVELOPMENT OF AN AUTOMATED PROCESS FOR TURBINE BLADE OPTIMISATION

Supervisor:
Prof. dr. sc. Hrvoje Jasak

Student:
Borna Šojat

Zagreb, 2017

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have fully cited all used sources and I have only used the ones given in the list of references

I would like to express my sincere gratitude to Professor Hrvoje Jasak for his guidance and encouragement whenever needed. His vast knowledge and valuable comments made this thesis possible.

I am also grateful to Tessa Uroić and Vanja Škurić for their immense support and help that greatly enhanced the quality of this thesis.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
procesno-energetski, konstrukcijski, brodostrojarski i inženjersko modeliranje i računalne simulacije

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

DIPLOMSKI ZADATAK

Student: **Borna Šojat** Mat. br.: 0035181628

Naslov rada na hrvatskom jeziku: **Razvoj automatiziranog procesa optimizacije profila turbinskih lopatica**

Naslov rada na engleskom jeziku: **Development of an automated process for turbine blade optimisation**

Opis zadatka:

Računalna dinamika fluida (*engl. Computational Fluid Dynamics*, CFD) primjenjuje se za optimizaciju komponenti turbostrojeva kako bi se smanjilo vrijeme potrebno za konstrukciju i proračun te u konačnici umanjila cijena gotovog proizvoda. Osobita je prednost primjene CFD-a što je u kratkom roku moguće ispitati velik broj varijacija konstrukcije, a što bi eksperimentalnim putem bilo nemoguće. U ovom radu razvit će se automatizirani proces optimizacije profila turbinskih lopatica. Proces će se sastojati od četiri koraka:

1. parametrizacije geometrije turbinskog profila,
2. deformacije početne proračunske mreže za novu geometriju,
3. numeričkog proračuna stlačivog strujanja kroz kanal statora,
4. odabira najboljeg rješenja prema zadanom kriteriju.

Kandidat će izvesti sljedeće zadatke tijekom izrade rada:

- odabrati i opisati algoritam za parametrizaciju te ga implementirati u obliku prikladnom za opisivanje geometrije profila turbinske lopatice,
- izraditi proračunsku mrežu za dvodimenzionalni statorski kanal, korištenjem paketa Pointwise,
- prilagoditi postojeći algoritam za deformaciju proračunske mreže u programskom paketu OpenFOAM za deformaciju mreže statorskog kanala,
- opisati jednadžbe te rubne uvjete koji se koriste za proračun stlačivih strujanja,
- odabrati prikladan algoritam optimizacije, dostupan u programskom paketu otvorenog koda Dakota te opisati karakteristike algoritma,
- povezati navedene korake procesa optimizacije skriptama kako bi se dobio potpuno automatizirani postupak,
- prikazati rezultate izvršene optimizacije te utvrditi mogućnost proširivanja procesa za primjenu na kompleksnijim geometrijama.

U radu navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:	Rok predaje rada:	Predviđeni datumi obrane:
17. studenog 2016.	19. siječnja 2017.	25., 26. i 27. siječnja 2017.

Zadatak zadao: Predsjednica Povjerenstva:

Prof. dr. sc. Hrvoje Jasak Prof. dr. sc. Tanja Jurčević Lulić

Table of Contents

Nomenclature

Greek letters

α	Closure coefficient	-
β	Closure coefficient	-
ε	Lift to drag ratio coefficient	-
γ	Diffusion coefficient	m^2 / s^2
κ	Adiabatic index	-
μ	Dynamic viscosity	Pa s
ω	Specific turbulence dissipation	1/s
ϕ	Transported scalar variable	-
ρ	Density	kg/ m^3
τ	Closure coefficient	-
ν_t	Kinematic eddy viscosity	m^2/s

Latin letters

A	Coefficient matrix	–
g	Gravitational acceleration	m / s^2
q	Heat flux	W· m
r	Right hand side vector	–
u	Velocity	m / s
A	Referent plane area	m^2
B_i	Position vector	-
c	Speed of sound	m/s
c_D	Drag coefficient	-
c_L	Lift coefficient	-

c_p	Heat capacity at constant pressure value	J/(kg·K)
c_v	Heat capacity at constant volume value	J/(kg·K)
$CD_{k\omega}$	Cross - diffusion in ω equation	-
e	Specific internal energy	J· m^3 / kg
F_1	Blending function	-
F_2	Blending function	-
F_D	Drag force	N
F_L	Lift force	N
I	Turbulent intensity	-
k	Turbulent kinetic energy	J/kg
M	Mach number	-
$N_{i,k}(t)$	Normalized basis function	-
P	Dynamic pressure	Pa
P_k	Blending function	-
p_{stat}	Static pressure	Pa
p_{tot}	Total pressure	Pa
Q	Volumetric heat source	W· m^3 / kg
T	Temperature	K
t	Time	s
T_{stat}	Static temperature	K
T_{tot}	Total temperature	K
y^+	Normalised distance to the wall	-

Abbreviations

CFD - Computation Fluid Dynamics

CPU - Central Processing Unit

NACA - National Advisory Committee for Aeronautics

RANS - Reynolds-Averaged Navier-Stokes Equations

SST - Shear Stress Transport

MOGA - Multi-Objective Genetic Algorithm

BC - Boundary Condition

EA - Evolutionary Algorithm

DAKOTA - Multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis

List of Figures

List of Tables

Abstract

Computational fluid dynamics (CFD) is a critical part in optimisation of turbomachinery components. Constant commercial pressure to produce parts of highest possible quality both in shortest possible time and with maximal reduction in price, are the reasons why CFD has become a necessity. Utilization of CFD enables faster design cycles, better performance, and reduction in cost and weight of the final product. CFD is of greatest importance in optimisation procedures because much more blade designs as well as blade parameters can be examined than it would be the case in an experimental study. Also CFD enables visualisation of complex fluid flow patterns over critical component geometries.
Developed automated optimisation process consists of the following steps:

1. Blade geometry parametrisation using a periodic B-spline curve,
2. Morphing of the computational mesh
3. Numerical calculation of a compressible fluid flow through turbine blade passage, and
4. Obtaining the new set of control polygon vertices.

In the parametrisation step, blade geometry, which is defined with the set of control polygon vertices, is approximated with a periodic B-spline curve of 5th degree. 40 parametrisation points are extrapolated between each pair of control polygon vertices. In the mesh morphing step, numerical mesh is deformed in respect with the parametrisation points coordinates. Numerical calculation of compressible fluid flow is conducted in the third step of this optimisation process in which objective function values are obtained. Objective functions are physical values derived from flow variables, which describe the characteristics of each blade geometry. Using this values, new set of control polygon vertices is calculated with Multi-objective evolutionary algorithm. In this way, the optimisation loop is closed and restarted. Due to the possibility of working with very irregular and distorted starting geometries, the developed automated optimisation process has proven to be robust and effective.

KEY WORDS: CFD, Optimisation, Turbomachinery, Evolutionary algorithm, Compressible transonic fluid flow

Sažetak

Računalna dinamika fluida (CFD, eng. Computational Fluid Dynamics) postala je neophodna u procesima optimizacije komponenti turbostrojeva. Smanjenje vremena potrebnog za konstrukciju i proračun rezultira nižom cijenom gotovog proizvoda. Brže pronalaženje optimalnog rješenja, a samim time i poboljšanje performansi te smanjenje mase gotovog proizvoda čine optimizaciju privlačnom za industrijsku upotrebu. Osim prethodno navedenih povoljnih svojstava računalne dinamike fluida, izrazito je bitno napomenuti mogućnost vizualizacije kompleksnih pojava u strujanju fluida oko kritičnih komponenti. U ovome je radu razvijen automatizirani proces optimizacije profila turbinskih lopatica. Proces se sastoji od sljedećih koraka:

1. Parametrizacija geometrije turbinskog profila,
2. Deformacija početne proračunske mreže za novu geometriju,
3. Numerički proračun stlačivog strujanja kroz kanal turbinske lopatice,
4. Odabir najboljeg rješenja prema zadanom kriteriju.

U prvom koraku se vrši postupak parametrizacije geometrije turbinske lopatice koristeći periodičnu *B-spline* krivulju 5 stupnja. Njome se aproksimira set kontrolnih točaka (kojima je definirana geometrija turbinske lopatice) te se između svake dvije kontrolne točke ekstrapolira 40 točaka parametrizacije. U drugom koraku se početna proračunska mreža deformira upravo prema koordinatama točaka parametrizacije da bi se dobila proračunska mreža oko traženog profila. U trećem koraku se vrši numerički proračun stlačivog strujanja kroz kanal turbinske lopatice te se izračunavaju vrijednosti funkcija cilja kao reprezentativne vrijednosti strujanja. Koristeći vrijednosti funkcija cilja, pomoću višeciljnog optimizacijskog algoritma, izračunava se novi set kontrolnih točaka. Time ponovo započinje petlja optimizacijskog procesa.

KLJUČNE RIJEČI: CFD, Optimizacija, Turbostrojevi, Evolucijski algoritam, Stlačivo transonično strujanje

Prošireni sažetak

(EXTENDED ABSTRACT IN CROATIAN)

Uvod

U ovome je radu predstavljen automatizirani proces optimizacije turbinske lopatice. Proces se sastoji od četiri dijela:

1. Parametrizacije geometrije turbinskog profila,
2. Deformacije početne proračunske mreže za novu geometriju,
3. Numeričkog proračuna stlačivog strujanja kroz kanal turbinske lopatice,
4. Odabir najboljeg rješenja prema zadanom kriteriju.

Struktura rada je podijeljena tako da se prvo predstavlja teorijski uvod za svaki korak optimizacijskog procesa, a zatim slijedi prezentacija i diskusija dobivenih rezultata.

Matematički model

Matematički model koji se koristi za modeliranje strujanja naziva se Navier-Stokesovim jednadžbama. Odnosno, za stlačivo strujanje to je sustav koji se sastoji od:

1. Jednadžbe kontinuiteta:
$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1}$$

2. Jednadžbe očuvanja količine gibanja:
$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \nabla \cdot [\mu (\nabla \mathbf{u} + (\nabla \mathbf{u}^T))] = \rho \mathbf{g} - \nabla (P + \frac{2}{3} \mu \nabla \cdot \mathbf{u}), \tag{2}$$

3. Jednadžbe očuvanja energije:
$$\begin{aligned} &\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) - \nabla \cdot (\lambda \nabla T) = \rho \mathbf{g} \cdot \mathbf{u} - \nabla \cdot (P\mathbf{u}) - \nabla \cdot (\frac{2}{3} \mu (\nabla \cdot \mathbf{u}) \mathbf{u}) \\ &+ \nabla \cdot [\mu (\nabla \mathbf{u} + (\nabla \mathbf{u}^T)) \cdot \mathbf{u}] + \rho Q. \end{aligned} \tag{3}$$

Da bi ovaj sustav jednadžbi bio rješiv potrebna je još jedna pomoćna jednadžba. U tu svrhu koristi se jednadžba stanja idealnog plina, koja predstavlja vezu gustoće ρ i tlaka p.

$$\rho = \frac{P}{RT}, \tag{4}$$

Paramerizacija

Parametrizacija geometrije turbinske lopatice provedena je koristeći periodičnu *B-spline* krivulju. Karakter bazne funkcije *B-spline* krivulje je generalno neglobalni, budući da svaka kontrolna točka B_i ima pripadajuću jedinstvenu baznu funkciju. Rekurzivnu definiciju *B-spline* krivulja otkrili su Cox [9] i de Boor [10] neovisno jedan o drugome, dok je Schoenberg [11] prvi iznio teoriju *B-spline* krivulja. Analitički zapisano, funkcija $P(t)$ glasi:

$$P(t) = \sum_{n=1}^{n+1} B_i N_{i,k}(t) \qquad t_{min} \leq t < t_{max}, \qquad 2 \leq k \leq n+1, \tag{5}$$

gdje je $N_{i,k}(t)$ normalizirana bazna funkcija. Daljnjom generalizacijom $i$ta normalizirana bazna funkcija $N_{i,k}(t)$ reda k je definirana Cox-de Boorovom rekurzivnom formulacijom:

$$N_{i,1}(t) = \begin{cases} 1 & \text{ako je} \quad x_i \leq t < x_{i+1} \\ 0 & \text{inače} \end{cases} \tag{6}$$

gdje je

$$N_{i,k}(t) = \frac{(t - x_i) N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t) N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}. \tag{7}$$

Vrijednosti x_i moraju biti monotoni niz rastućih realnih brojeva, gdje je zadovoljena relacija $x_i \leq x_{i+1}$. Dvije karakteristike *B-spline* krivulja su bile presudne za njezin izbor u parametrizacijskom postupku. Ova vrsta krivulja ne interpolira kontrolne točke kojima je zadana geometrija, nego ih aproksimira što je bolje moguće. Na taj su način izbjegnuti diskontinuiteti na geometriji, tijekom manipulacije geometrije, gdje je korištena promjena pozicija kontrolnih točaka (drugo bitno svojstvo). Budući da je trebalo opisati turbinsku lopaticu, odabrana je periodična *B-spline* krivulja 5-og stupnja zbog činjenice da posjeduje povoljna svojstva za opisivanje zatvorenih krivulja [3].

1. Utjecaj bazne funkcije je limitiran na k intervala, i

2. Sve *B-spline* bazne funkcije su translacije iste funkcije, odnosno na intervalu $0 \leq t^* < 1$ sve imaju formu $N^*_{i,k}(t^*)$ [3]. Reparametrizacija bazne funkcije na dani interval točka

računa se kao:

$$P_j(t^*) = \sum_{i=0}^{k-1} N^*_{i+1,k}(t^*) B_{j+i} \qquad 1 \le j \le n-k+1, \quad 0 \le t^* < 1 \tag{8}$$

gdje je j broj segmenata krivulje, a n je jedan manje od broja kontrolnih točaka.

Periodična *B-spline* funkcija se može zapisati i u matričnoj formi te glasi:

$$P_j(t^*) = [F][G] \qquad 1 \le j \le n-k+1, \qquad 0 \le t^* < 1 \tag{9}$$

odnosno, to je produkt matrice *blending* funkcija [F] i geometrijske matrice [G] koja sadrži koordinate kontrolnih točaka.

$$[G] = \begin{bmatrix} B_j \\ . \\ . \\ . \\ B_{j+k-1}, \end{bmatrix} \tag{10}$$

Matrica $[F]$ se dalje raspisuje kao:

$$[F] = [T^*][N^*] \tag{11}$$

gdje je $[T^*]$

$$[T^*] = [t^{*k-1} \quad t^{*k-2} \quad \cdots \quad t^* \quad 1] \qquad 0 \le t^* < 1 \tag{12}$$

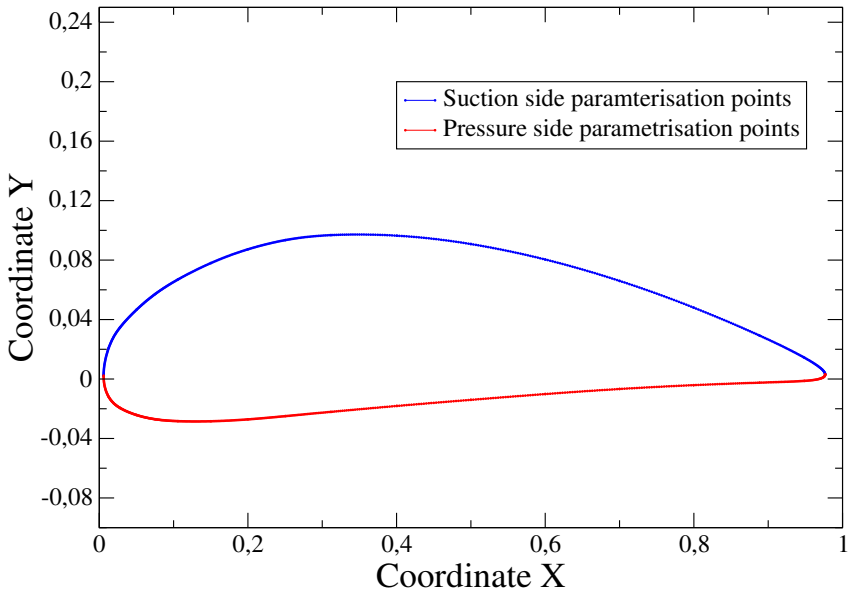
što zapravo predstavlja matrični zapis ekstrapoliranih parametrizacijskih točaka između dvije kontrolne točke. Cohen i Riesenfeld [12] su predložili generaliziranu formu $[N]$ kao:

$$[N^*] = [N^*_{i+1,j+1}] \tag{13}$$

gdje je $N^*_{i+1,j+1}$:

$$N^*_{i+1,j+1} = \frac{1}{(k-1)!} \binom{k-1}{i} \sum_{l=j}^{k-1} (k-(l+1))^i (-1)^{l-j} \binom{k}{l-j} \qquad 0 \le i, j \le k-1 \tag{14}$$

Korištenjem navedenih izraza izračunavaju se točke parametrizacije koje opisuju geometriju turbinske lopatice te se prema njima u drugom koraku vrši deformacija proračunske mreže. Ove točke su prikazane na Sl. 1



Slika 1: Prikaz točaka parametrizacije.

Deformacija proračunske mreže

U svrhu pronalaska optimalne (globalno rješenje) geometrije turbinske lopatice, potrebno je provesti velik broj iteracija optimizacijskog algoritma, a samim time i generirati velik broj proračunskih mreža. U [14] je dan pregled numeričkih grešaka koje nastaju kao: posljedica diskretizacije konvektivnih i vremenskih članova, te greške povezane s proračunskom mrežom. Budući da je u ovom procesu osigurana konzistentna upotreba diskretizacijskih shema, glavni potencijalni izvor greške bila je proračunska mreža. Iz tog je razloga ujednačena kvaliteta i lokalna rezolucija mreže iznimno važna. Ručnim generiranjem proračunske mreže moguće je zadovoljiti navedene kriterije, ali je vremenski zahtjevno. S druge pak strane, korištenje automatski generiranih mreža je povoljno u vidu potrebnog vremena za njihovo generiranje, ali je kontrola lokalne rezolucije onemogućena. Iz tog je razloga korišteno deformiranje proračunske mreže pomicanjem njezinih točaka. Jednaka kvaliteta i lokalna rezolucije proračunske mreže je osigurana koristeći istu početnu (inicijalnu) mrežu u svakoj iteraciji optimizacijskog procesa. Ova procedura je bazirana na pretpostavci da se proračunska mreža ponaša kao elastično tijelo u stanju ravnoteže [15]. U svrhu pomicanja površine mreže primijenjena je Laplace-ova jednadžba:

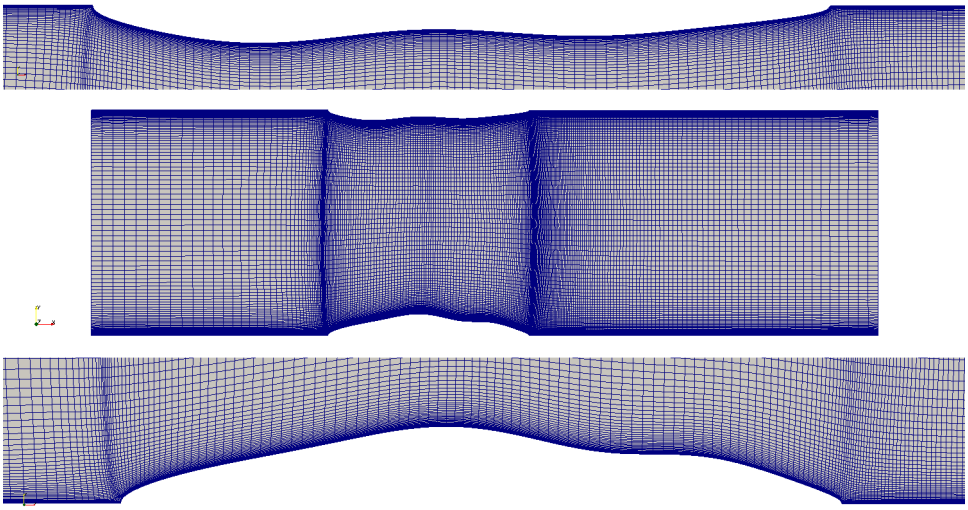
$$\nabla \cdot (\gamma \nabla \mathbf{u}) = 0 \tag{15}$$

gdje je $\mathbf{u}$ vektor pomaka čvorova, a γ koeficijent difuzije. Ukoliko se koristi jednadžba (15) s konstantnim koeficijentom γ, najveće deformacije proračunske mreže pojavit će se uz granice. To dovodi do povećanja mogućnosti nastanka ćelija s negativnim volumenima. Iz tog razloga

je u izradi ovog rada korišten prostorno ovisan koeficijent difuzije koji je osigurao znatno nižu distorziju proračunske mreže. On može biti dan kao linearna, kvadratična ili eksponencijalna funkcija.

$$\gamma(l) = l^{-2}, \tag{16}$$

U ovom radu korištena je kvadratična forma, jednadžba (16). Tražena geometrija proračunske mreže se definira setom točaka koje su uvedene u jednadžbi (15) kao Dirichletov rubni uvjet. Na Sl. 2 je prikazana inicijalna proračunska mreža. Deformacija mreže se odvija u dvije iteracije. U prvoj iteraciji se proračunska mreža modificira u aksijalnom smjeru prema danim



Slika 2: Inicijalna proračunska mreža turbinskog kanala.

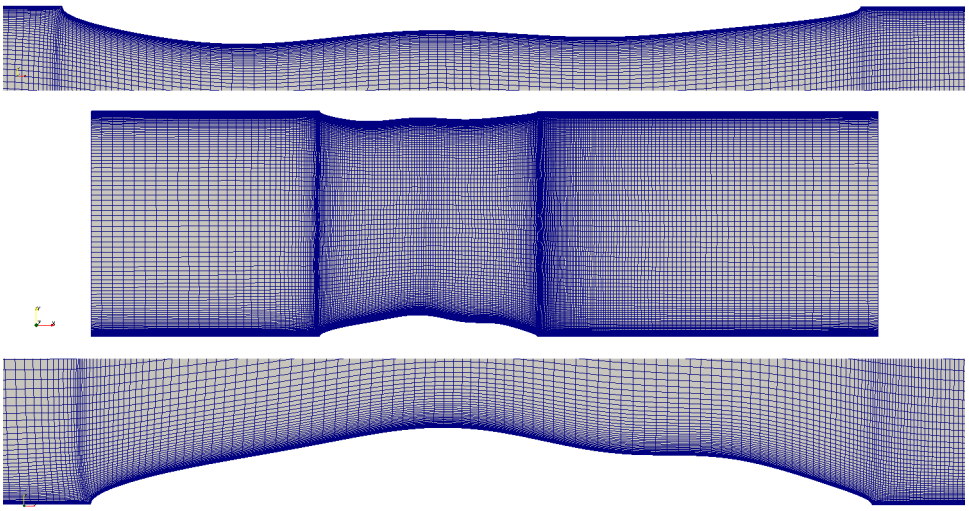
koordinatama točaka parametrizacije iz prvog koraka optimizacijskog procesa. Korišteni rubni uvjeti u prvoj iteraciji deformacije mreže su prikazani u Tablici 1, a izgled proračunske mreže na Sl. 3.

Table 1: Rubni uvjeti korišteni za deformaciju proračunske mreže u aksijalnom smjeru.

Patch	Boundary condition
Inlet	slip
Outlet	slip
blade_ps	Dirichlet
periodic	slip
blade_ss	Dirichlet
shadow	slip
Front	empty
Back	empty

U drugoj iteraciji se vrši deformacija proračunske mreže u tangencijalnom smjeru prema koordinatama točaka parametrizacije. Rubni uvjeti korišteni u drugoj iteraciji prikazani su u

Tablici 2.

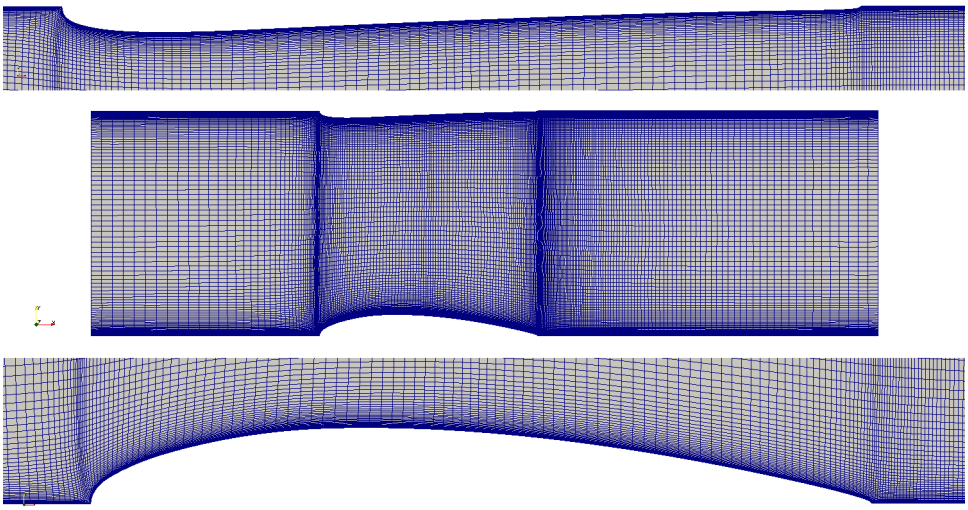


Slika 3: Izgled proračunske mreže turbinskog kanala nakon prve iteracije deformiranja mreže.

Table 2: Rubni uvjeti korišteni za deformaciju proračunske mreže u tangencijalnom smjeru.

Patch	Boundary condition
Inlet	slip
Outlet	slip
blade_ps	Dirichlet
periodic	Dirichlet
blade_ss	Dirichlet
shadow	Dirichlet
Front	empty
Back	empty

Proračunska mreža dobivena u drugoj iteraciji, odnosno nakon deformacije u tangencijalnom smjeru, prikazana je na Sl. 4. Nakon što je inicijalna proračunska mreža modificirana prema danim točkama parametrizacije, provodi se sljedeći korak optimizacijskog procesa, numerički proračun stlačivog strujanja.



Slika 4: Proračunska mreža dobivena nakon druge iteracije deformiranja mreže.

Numerički proračun stlačivog strujanja

Rubni uvjeti korišteni za numerički proračun stlačivog strujanja kroz turbinski kanal su predstavljeni u Tablici 3.

Table 3: Rubni uvjeti korišteni za numerički proračun stlačivog strujanja.

Patch	**Velocity BC**	**Pressure BC**	**Temperature BC**
Inlet	`pressureDirectedInletVelocity`	`isentropicTotalPressure`	`isentropicTotalTemperature`
Outlet	`inletOutlet`	Dirichlet	Neumann
blade_ps	Dirichlet	Neumann	Neumann
periodic	`cyclicGGI`	`cyclicGGI`	`cyclicGGI`
blade_ss	Dirichlet	Neumann	Neumann
shadow	`cyclicGGI`	`cyclicGGI`	`cyclicGGI`
Front	empty	empty	empty
Back	empty	empty	empty

Budući da se u ovom optimizacijskog procesu traži optimalna geometrija turbinskih lopatica u transoničnom kompresibilnom strujanju fluida, posebna pažnja je posvećena postavljanju rubnih uvjeta. Za subsonično i transonično stlačivo strujanje na izlazu iz proračunske domene se definira jedna varijabla, dok se na ulazu u proračunsku domenu definiraju dvije. U optimizacijskom procesu rotorske lopatice vrijednost statičkog tlaka na izlazu iz domene je postavljena na 100 000 Pa, dok su vrijednosti dvije varijable postavljene na ulaz u proračunsku domenu izračunate jednadžbom (17) i jednadžbom (18). Ove vrijednosti su dobivene za Machov broj $M = 0.6$ i statičku temperatura $T = 288.15$ K na izlazu iz domene.

$$\frac{p_{tot}}{p_{stat}} = \left(1 + \frac{\kappa - 1}{2} M^2\right)^{\frac{\kappa}{\kappa - 1}}, \tag{17}$$

$$\frac{T_{tot}}{T_{stat}} = 1 + \frac{\kappa - 1}{2} M^2, \tag{18}$$

gdje je p_{tot} totalni tlak, p_{stat} statički tlak, T_{tot} totalna temperatura i T_{stat} statička temperatura.

U procesu optimizacije statorske lopatice statički tlak na izlazu iz domene je postavljen na 150 000 Pa, a vrijednosti totalnog tlaka i totalne temperature na ulazu u domenu su izračunate prema jednadžbi (17) i jednadžbi (18). Vrijednost Machovog broja na izlazu iz domene je bila postavljena, kao i u slučaju za rotorsku lopaticu, na $M = 0.6$, dok je statička temperatura na izlazu iznosila 350 K.

`pressureDirectedInletVelocity` je korišten kao rubni uvjet za brzinu na ulazu u proračunsku domenu jer omogućava variranje napadnog kuta pri kojem fluid nastrujava na lopaticu. Vrijednost brzine na lopaticama iznosi 0 m/s dok su vrijednosti tlaka i temperature na lopaticama izračunate pomoću Neumannovog rubnog uvjeta.

Višeciljni genetski optimizacijski algoritam

U četvrtom koraku automatiziranog optimizacijskog procesa vrši se proračun novih kontrolnih točaka kojima je opisana geometrija turbinske lopatice. U tu je svrhu korištena MOGA (eng. Multi-Objective Genetic Algorithm) koja je sastavni dio DAKOTA-e, softverskog paketa otvorenog koda. MOGA funkcionira izvršavajući sljedeće korake:

1. Inicijalizacija populacije
   Početna (inicijalna) populacija je nasumično generirana, te svaka jedinka ima svoj vlastiti set kontrolnih točaka, odnosno svoj vlastiti "genetski kod".

2. Evaluacija jedinki inicijalne populacije
   Za svaku jedinku iz inicijalne populacije izračunavaju se vrijednosti funkcija cilja. Također, za svaku jedinku iz populacije se izračunavaju ograničenja te se ocjenjuje jesu li ona zadovoljena.

3. Križanje jedinki
   Iz inicijalne populacije jedinki, odabrani su parovi za razmnožavanje (križanje). Svaki par može proizvesti zadani broj potomaka

4. Mutacija jedinki
   Na nasumično izabranom broju potomaka vrši se mutacija

5. Evaluacija novih jedinki

6. Ocjena dominantnosti jedinki u populaciji

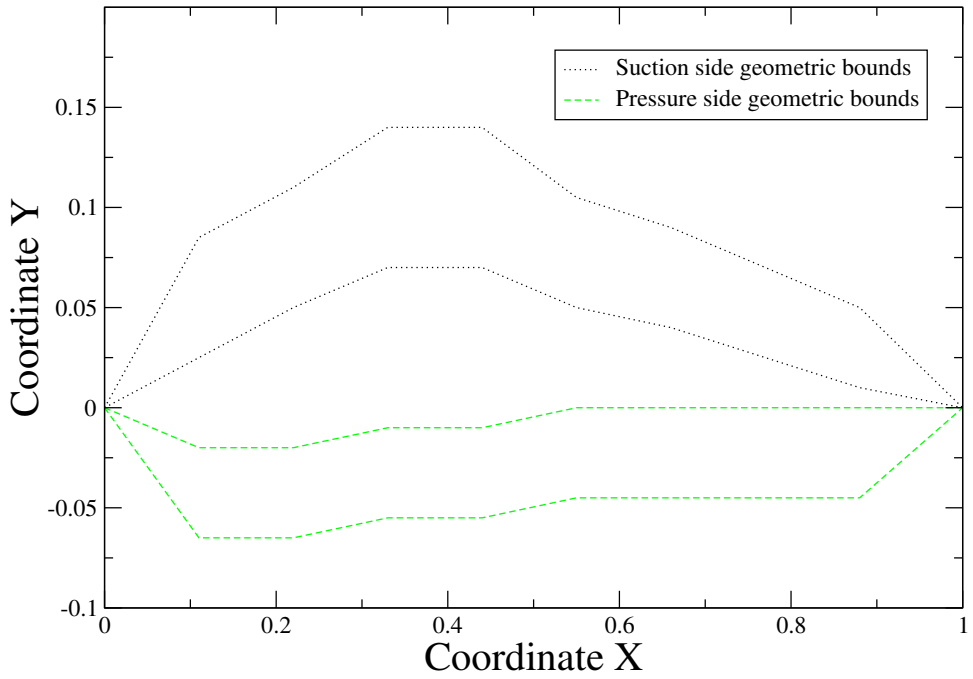
7. Uvođenje potomaka u populaciju

   Svi potomci, koji su prethodno rangirani prema dominantnosti, se uspoređuju s jedinkama u populaciji. Ukoliko se pokaže da je neki od potomaka dominantniji od najnedominantnije jedinke u populaciji, vrši se uvođenje tog potomka u populaciju. Ukoliko se pokaže da niti jedan od potomaka nije dominantniji, njihov ulazak u populaciju je zabranjen.

8. Novi odabir jedinki za razmnožavanje

   Zadaje se određeni broj parova za razmnožavanje, a za njega se razmatraju sve jedinke. Budući da se dominantnije jedinke multipliciraju u popisu za razmnožavanje, vjerojatnost za njihov odabir je povećana.

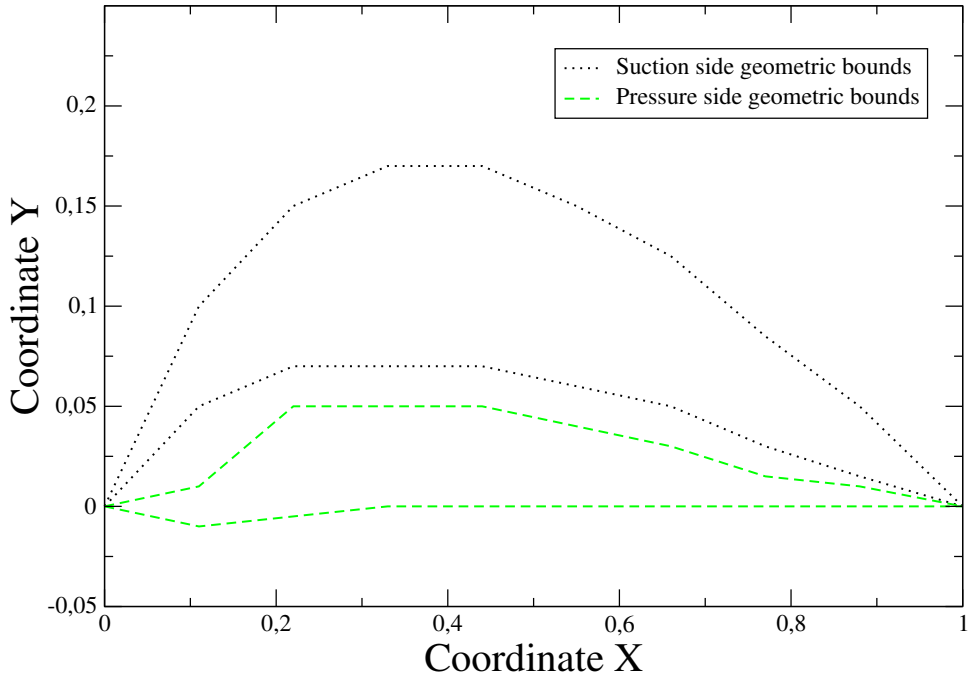
Ovaj iterativni proces se ponavlja od koraka 4 do koraka 8 dok jedan od zadanih kriterija optimizacije nije ispunjen.



Slika 5: Geometrijska ograničenja za optimizaciju rotorske lopatice.

Maksimalni broj evaluacija funkcija je bio 10 000, ali je to daleko više od broja koji se ostvaruje tijekom optimizacije. Tako da je kriterij optimizacije bio maksimalni broj generacija populacije ili dovoljno mala promjena vrijednosti funkcije cilja u populaciji. Ono što je bitno napomenuti, geometrijska ograničenja za optimizaciju kako rotorske tako i statorske lopatice su pomno odabrana, te su prikazani na Sl. 5 i Sl. 6. Potrebno je postaviti dovoljno jaka geometrijska ograničenja, kako bi se smanjio potreban broj iteracija. S druge strane, loše postavljena geometrijska ograničenja mogu dovesti do konvergencije u suboptimalno rješenje.

Najbolji primjer za to je činjenica da ravna ploča pri odgovarajućem napadnom kutu može imati vrijednost omjera koeficijenta uzgona i otpora jadnaku 10.



Slika 6: Geometrijska ograničenja za optimizaciju statorske lopatice.

Rezultati optimizacijskog procesa rotorske lopatice pri napadnom kutu od 7.5^o

U turbinama se snaga proizvodi pretvarajući tlak medija u koristan mehanički rad. Odnosno, koristeći rotacijski element (rotor) pokušava se iz struje fluida preuzeti što više energije. Iz tog su razloga kao funkcije cilja u optimizaciji rotorske lopatice korišteni koeficijent otpora c_D (minimizírati vrijednosti) i omjer koeficijenta uzgona i otpora ε (maksimírati vrijednosti). Koeficijent otpora c_D se izračunava prema:

$$c_D = \frac{F_D}{\frac{1}{2}\rho u^2 A}, \tag{19}$$

gdje je F_D sila otpora, a A je referentna površina lopatice izračunata kao produkt visine lopatice i duljine tetive. Koeficijent uzgona se računa kao:

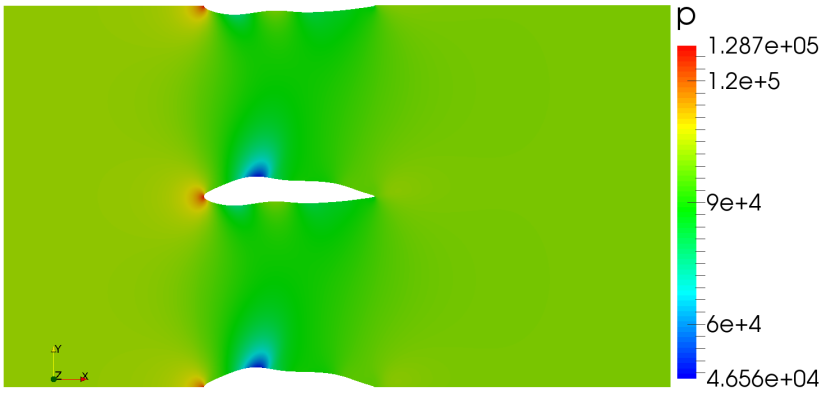
$$c_L = \frac{F_L}{\frac{1}{2}\rho u^2 A}; \tag{20}$$

gdje je F_L sila uzgona, a A je, kao što je to slučaj za koeficijent otpora, referentna površina lopatice.

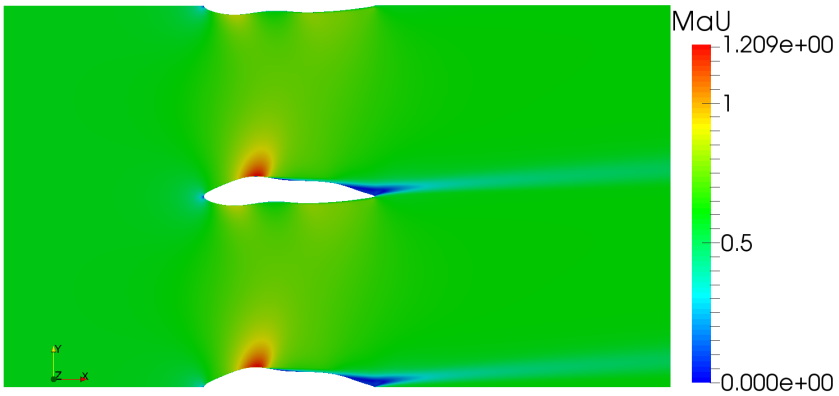
ε se definira kao:

$$\varepsilon = \frac{c_L}{c_D} \tag{21}$$

Prvo će biti prikazani rezultati dobiveni numeričkim proračunom stlačivog strujanja kroz kanal inicijalne geometrije turbinske lopatice, a zatim kroz kanal optimirane geometrije.

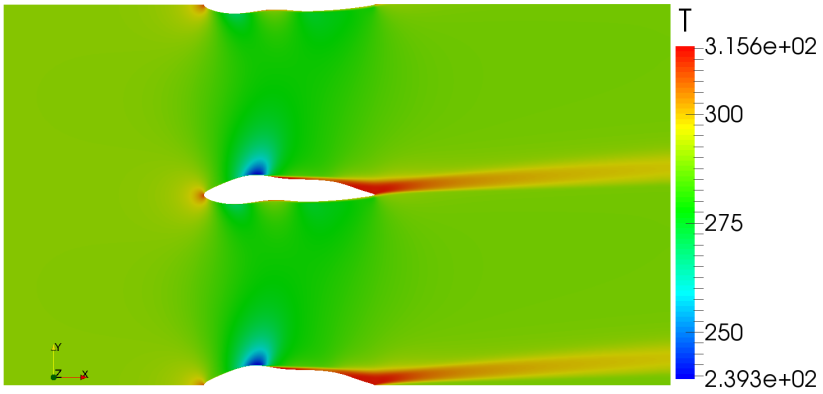


Slika 7: Polje tlaka inicijalne geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o.

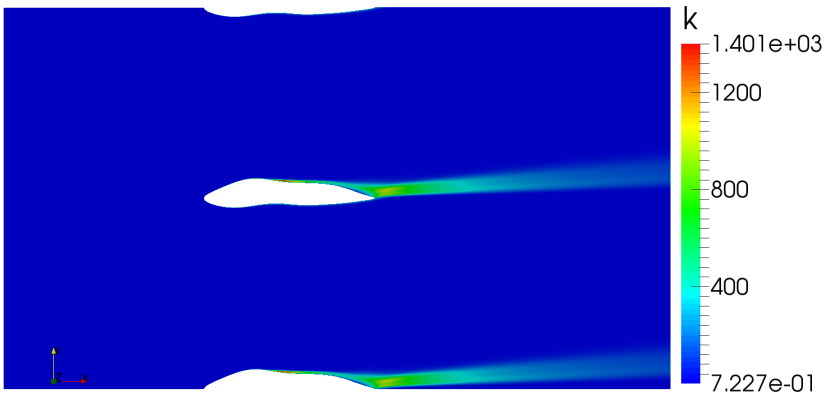


Slika 8: Polje Machovog broja inicijalne geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o.

Na Sl. 7, Sl. 8 i Sl. 9 prikazana su redom polja tlaka, Machovog broja i temperature. Na Sl. 7 i Sl. 9 mogu se zamijetiti lokalni minimumi tlaka i temperature na pretlačnoj i potlačnoj strani lopatice. Razlog za to je neaerodinamičnost inicijalne geometrije turbinske lopatice. Na mjestima lokaliziranog pada tlaka i temperature dolazi do lokaliziranog povećanja brzine radnog fluida. Razlog za to pronalazimo u Bernoullijevoj jednadžbi. Iz Sl. 8 vidi se da je strujanje radnog fluida u transoničnom području.

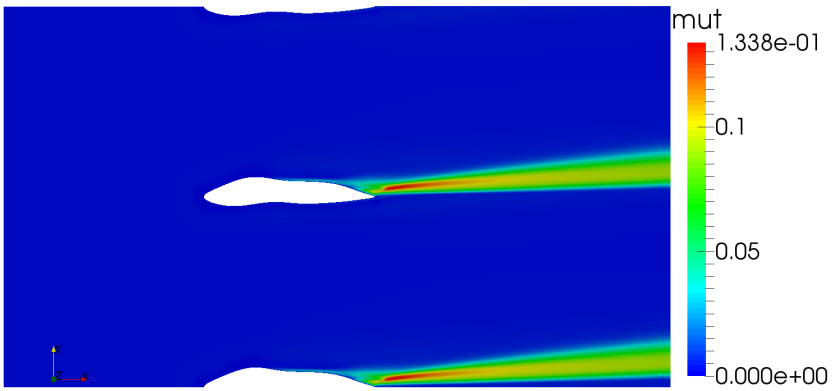


Slika 9: Temperaturno polje inicijalne geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o.



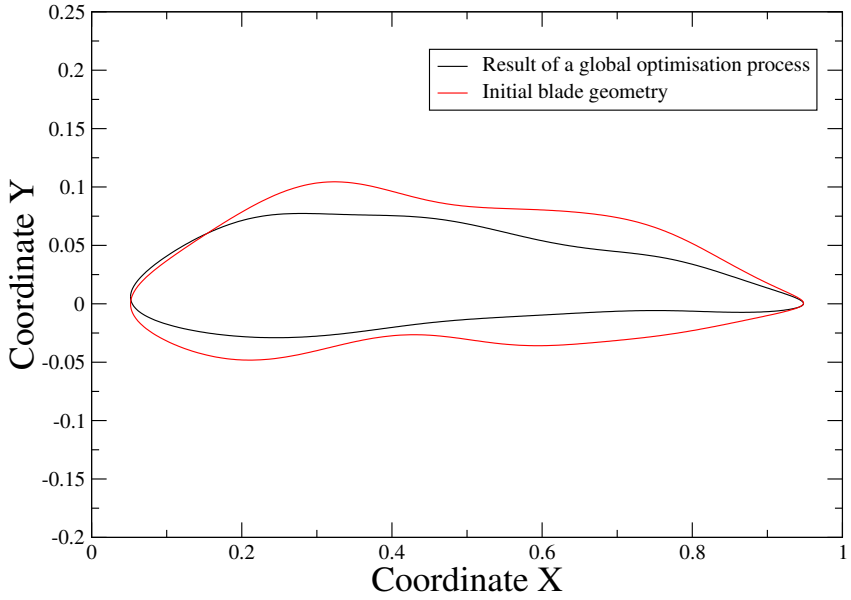
Slika 10: Polje turbulentne kinetičke energije inicijalne geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o.

Na Sl. 10 i Sl. 11 prikazana su polja turbulentne kinetičke energije i turbulentne viskoznosti pri vrijednosti kuta nastrujavanja fluida 7.5^o. Iz ovih polja jasno je vidljivo da je vrtložni trag ove lopatice usmjeren prema gore, odnosno u smjeru strujanja fluida. Također, lokalne vrijednosti ovih polja su relativno visoke. Budući da ova lopatica nije zaokrenula smjer strujanja fluida, za očekivati je nisku vrijednost sile uzgona.



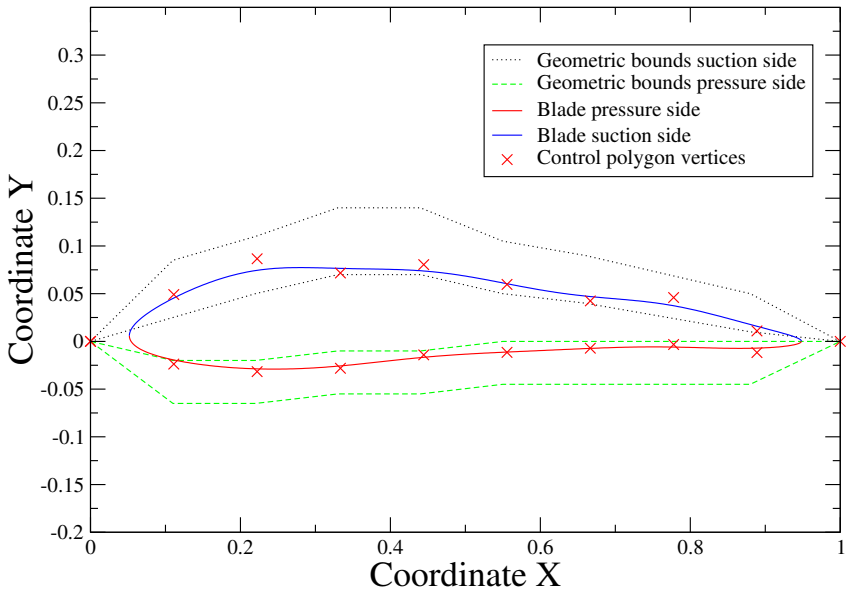
Slika 11: Polje turbulentne viskoznosti inicijalne geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o.

Na Sl. 12 prikazana je usporedba inicijalne i optimirane geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o. Može se vidjeti da je automatizirani optimizacijski proces pronašao globalni optimum koji podsjeća na NACA aeroprofil, kao što je i očekivano.

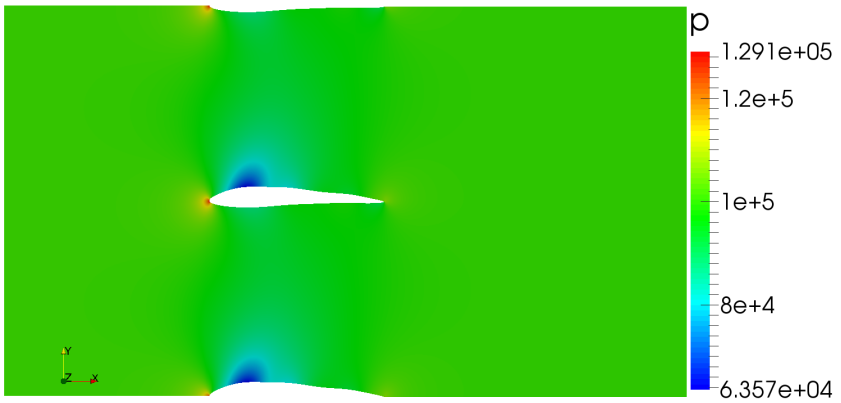


Slika 12: Usporedba inicijalne i optimirane geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o.

Na Sl. 13 prikazana je optimirana geometrija rotorske lopatice pri kutu nastrujavanja fluida 7.5^o. Također, mogu se vidjeti kontrolne točke kojima je definiran geometrija turbinske lopatice kao i njihovu poziciju u odnosu na geometrijska ograničenja. Moguće je zamijetiti bitno svojstvo *B-spline* krivulja, a to je da nema diskontinuiteta na geometriji.

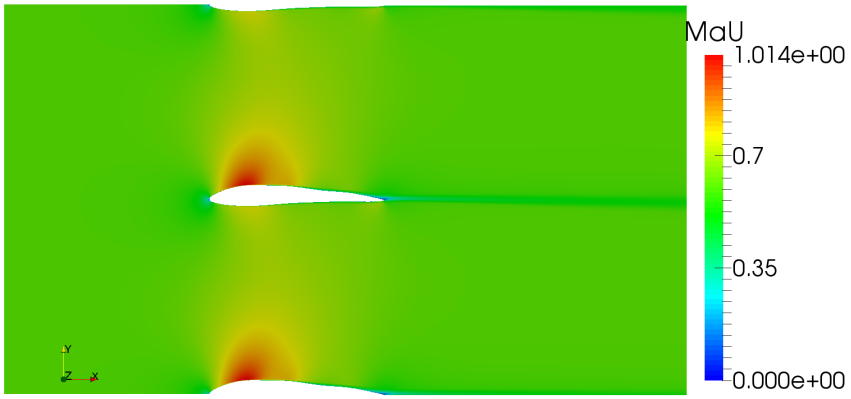


Slika 13: Prikaz optimirane geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o.

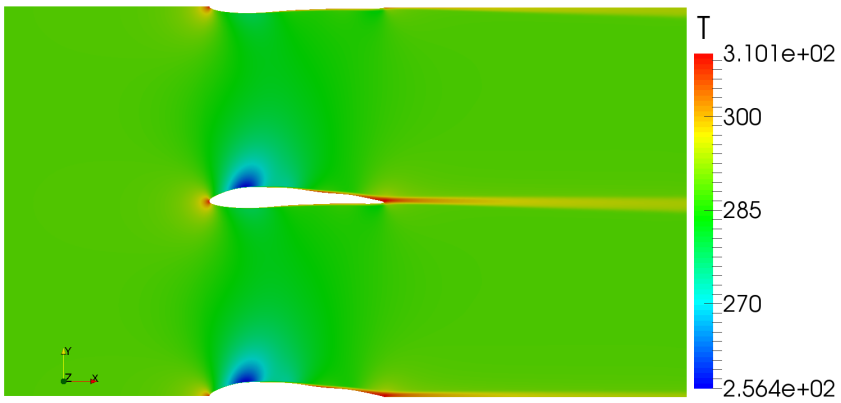


Slika 14: Polje tlaka optimirane geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o.

Na Sl. 14, Sl. 15 i Sl. 16 su prikazana polja: tlaka, Machovog broja i temperature. Sa slike Sl. 14 i Sl. 15 može se vidjeti da postoji pad tlaka isključivo na potlačnoj strani lopatice što dovodi do zaključka da je ova rotorska lopatica sposobna proizvesti znatnu silu uzgona. Prema Bernoullijevoj jednadžbi možemo zaključiti da će na istom mjestu doći do ubrzanja radnog fluida. To se može vidjeti na Sl. 15 kao i činjenica da je strujanje u potpunosti u transoničnom području.

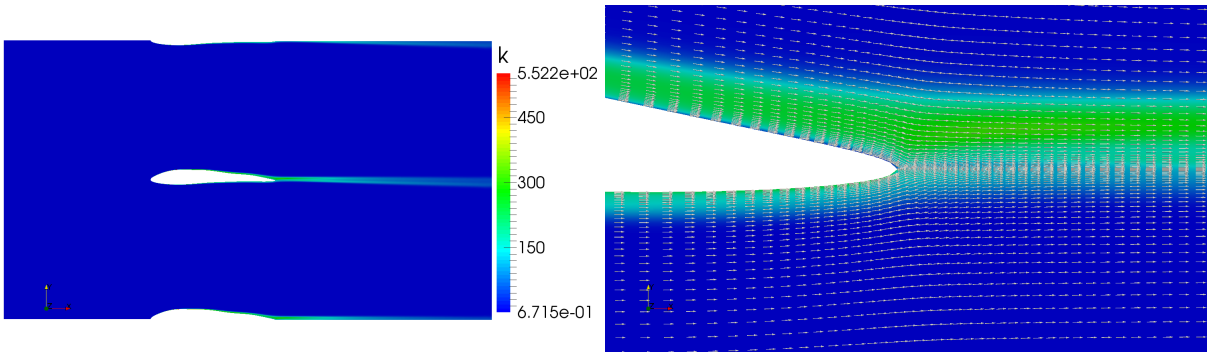


Slika 15: Polje Machovog broja optimirane geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o.

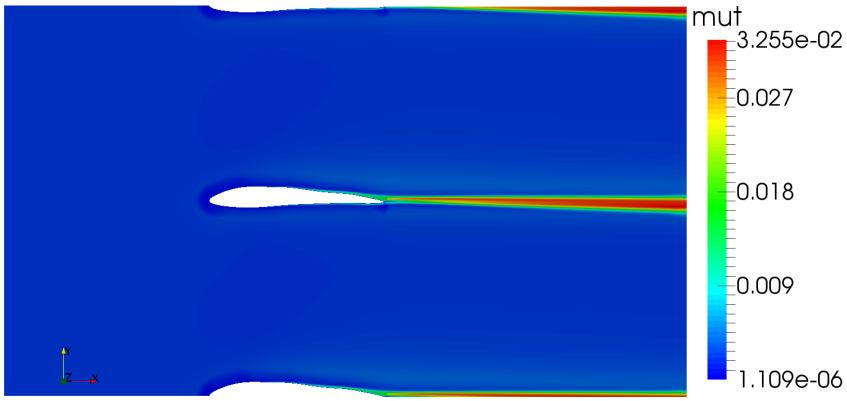


Slika 16: Temperaturno polje optimirane geometrije rotorske lopatice pri kutu nastrujavanja fluida 7.5^o.

Na Sl. 17 prikazano je polje turbulentne kinetičke energije optimirane geometrije rotorske lopatice s detaljom izlaznog brida pri kutu nastrujavanja fluida 7.5^o, dok je na Sl. 18 prikazano polje turbulentne viskoznosti. U usporedi s vrijednostima izračunatim za inicijalnu geometriju, može se vidjeti da je maksimalna vrijednost ovih polja značajno niža. Također i na Sl. 17 i na Sl. 18 vidi se vrtložni trag iza rotorske lopatice. Iako je nastrujavanje fluida na lopaticu pri kutu 7.5^o, vrtložni trag je usmjeren prema dolje. Takav zakret u strujanju radnog fluida također pridonosi zaključku da ova lopatica generira značajnu silu uzgona.



Slika 17: Polje turbulentne kinetičke energije optimirane geometrije rotorske lopatice s detaljom izlaznog brida pri kutu nastrujavanja fluida 7.5^o.



Slika 18: Polje turbulentne viskoznosti optimirane geometrije rotorske lopatice pri kutu nastrujavanja fluida od 7.5^o.

Za pronalazak ovog optimalnog rješenja, automatizirani optimizacijski proces izvršio je 77 iteracija, dok je višeciljni genetski optimizacijski algoritam inicijalizirao 12 generacija. U Tablici 4, prikazani su rezultati funkcija cilja za inicijalnu i optimiranu geometriju.

Table 4: Usporedba vrijednosti funkcija cilja inicijalne i optimirane geometrije rotorske lopatice.

Funkcije cilja	Inicijalna geometrija	Optimirana geometrija
ε	2.564	37.264
c_D	0.0754	0.01243

Rezultati optimizacijskog procesa statorske lopatice pri napadnom kutu od 5^o

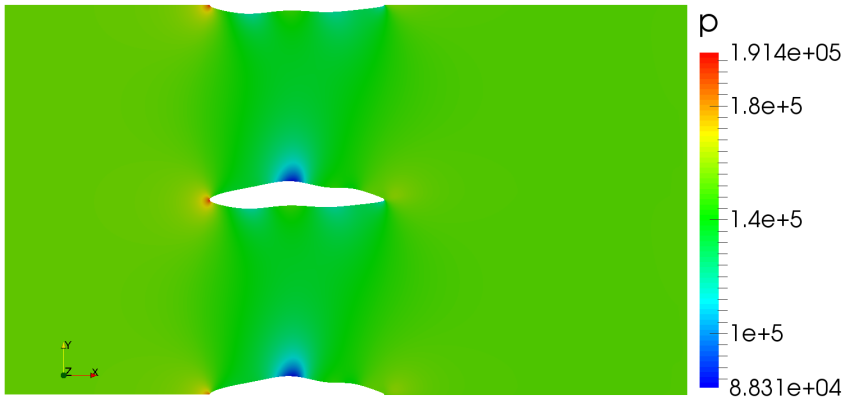
U turbinama se statorske lopatice koriste za što veće ubrzavanje struje radnog fluida pri što manjem padu tlaka. Također, statorske lopatice usmjeravaju struju fluida na rotorske lopatice

pri što povoljnijem napadnom kutu. Imajući to na umu, definirane su funkcije cilja za optimizaciju statorske lopatice, odnosno Δu, i Δp. Ove vrijednosti se izračunavaju kao:

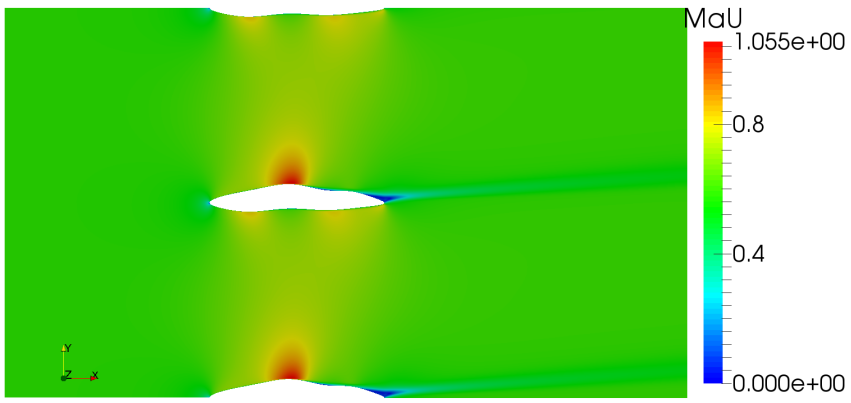
$$\Delta u = u_{outlet} - u_{inlet}, \tag{22}$$

$$\Delta p = p_{inlet} - p_{outlet}. \tag{23}$$

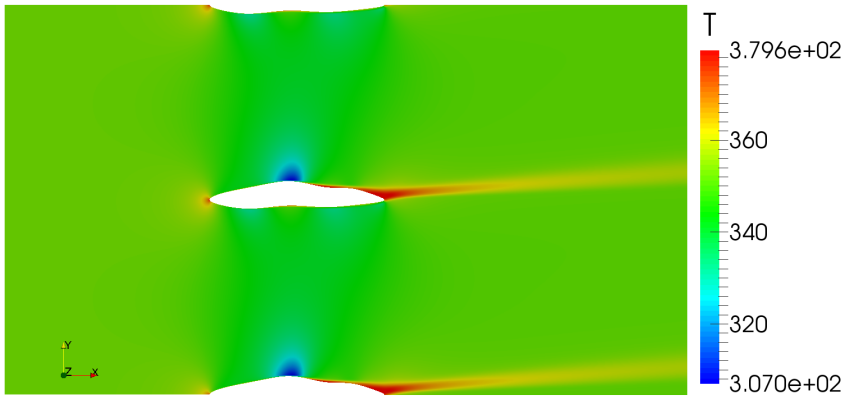
U nastavku će prvo biti prikazani rezultati za inicijalnu geometriju, a zatim za optimiranu.



Slika 19: Polje tlaka inicijalne geometrije statorske lopatice pri kuta nastrujavanja fluida od 5^o.

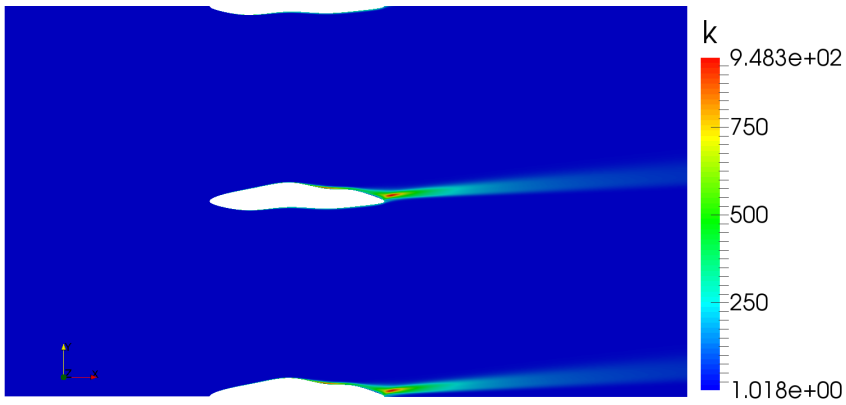


Slika 20: Polje Machovog broja za inicijalnu geometriju statorske lopatice pri kuta nastrujavanja fluida od 5^o.

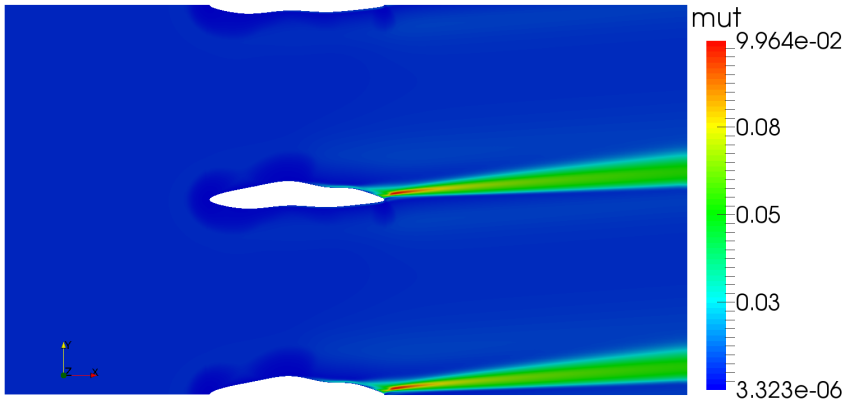


Slika 21: Temperaturno polje inicijalne geometrije statorske lopatice pri kuta nastrujavanja fluida od 5^o.

Na Sl. 19, Sl. 20 i Sl. 21 prikazana su redom polja tlaka, Machovog broja i temperature izračunata za stlačivo strujanje fluida kroz turbinski kanal inicijalne geometrije. Sl. 20 jasno prikazuje da se strujanje radnog fluida nalazi u transoničnom području. Također, jasno su vidljivi i lokalizirani maksimumi brzine radnog fluida na pretlačnoj i potlačnoj strani. Sa Sl. 19 i Sl. 21 vidljivi su lokalizirani minimumi tlaka i temperature na istim mjestima. Na Sl. 22 prikazano je polje turbulentne kinetičke energije inicijalne geometrije statorske lopatice pri kuta nastrujavanja fluida 5^o, dok je na Sl. 23 prikazano polje turbulentne viskoznosti. Na obje slike je vidljivo da je vrtložni trag usmjeren prema gore. Maksimalna vrijednost turbulentne kinetičke energije je lokalizirana uz izlazni brid inicijalne geometrije.

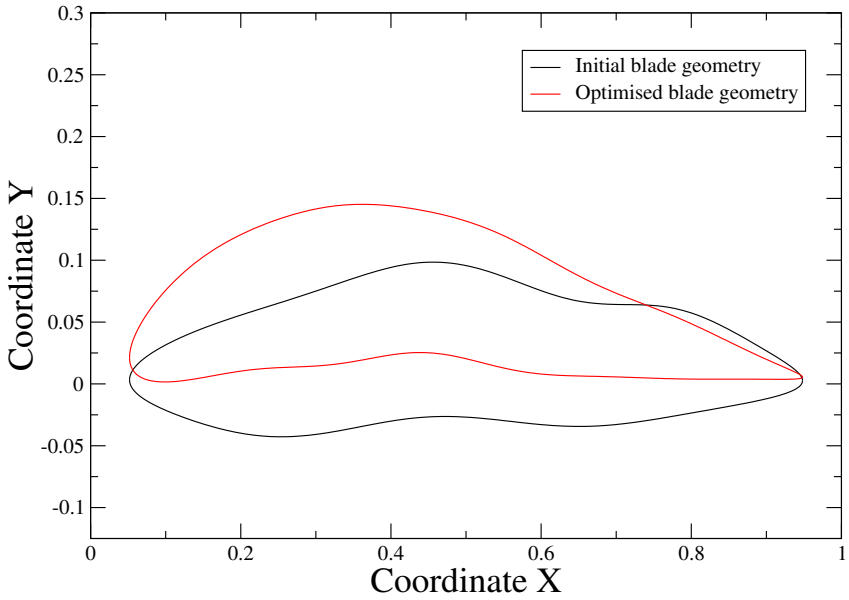


Slika 22: Polje turbulentne kinetičke energije inicijalne geometrije statorske lopatice pri kuta nastrujavanja fluida od 5^o.

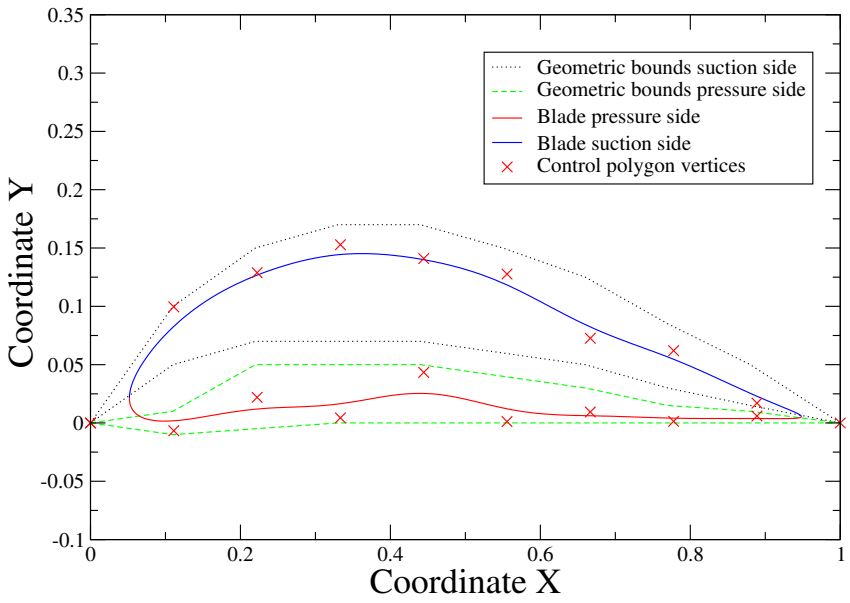


Slika 23: Polje turbulentne viskoznosti inicijalne geometrije statorske lopatice pri kuta nastrujavanja fluida od 5^o.

Na Sl. 24 prikazana je usporedba inicijalne i optimirane geometrije statorske lopatice pri vrijednosti kuta nastrujavanja fluida od 5^o, dok je na Sl. 25 prikazana optimirana geometrija statorske lopatice s definirajućim kontrolnim točkama. Također, na slici su prikazane i geometrijske granice u kojima se moraju nalaziti kontrolne točke.

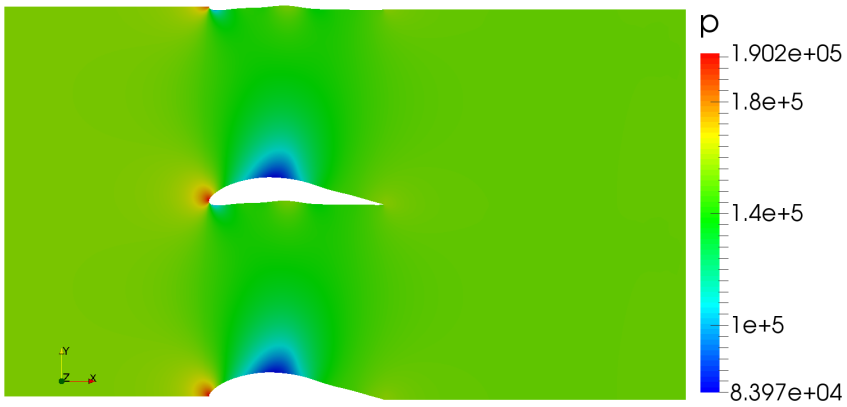


Slika 24: Usporedba inicijalne i optimirane geometrije statorske lopatice pri kuta nastrujavanja fluida od 5^o.

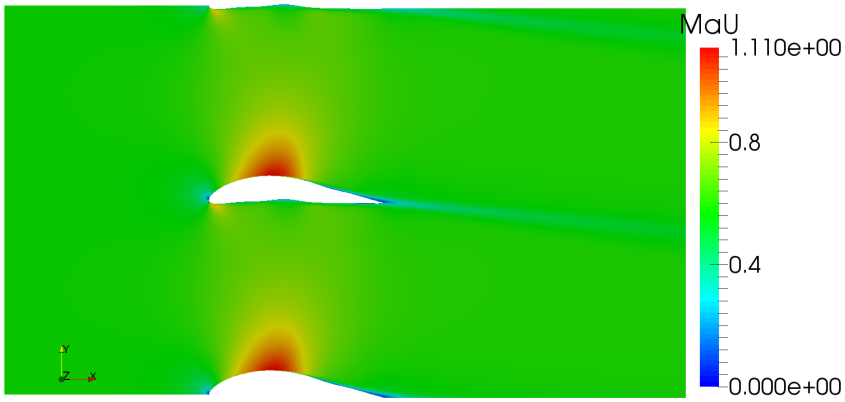


Slika 25: Optimirana geometrija statorske lopatice pri kuta nastrujavanja fluida od 5^o.

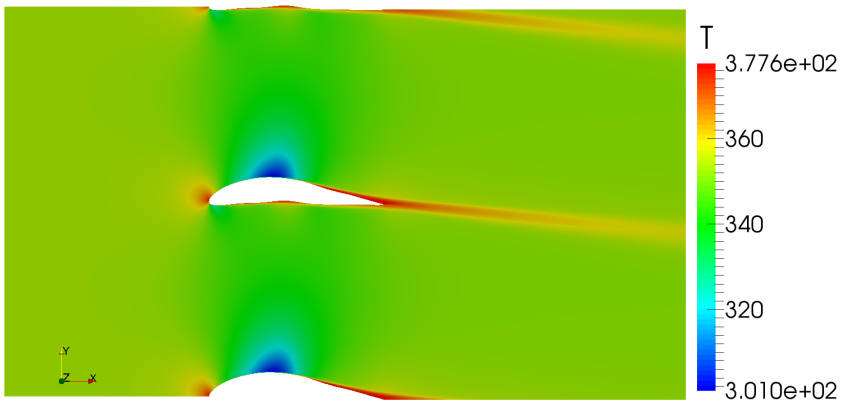
Sl. 26, Sl. 27 i Sl. 28 prikazuju polja tlaka, Machovog broja i temperature. Ove vrijednosti su dobivene numeričkim proračunom stlačivog strujanja kroz kanal optimirane statorske lopatice. Sa Sl. 27 vidljivo je da se radni fluid nalazi u transoničnom strujanju. Također, pad tlaka i temperature je koncentriran na potlačnoj strani lopatice, gdje prema Bernoullijevoj jednadžbi dolazi i do ubrzanja radnog fluida.



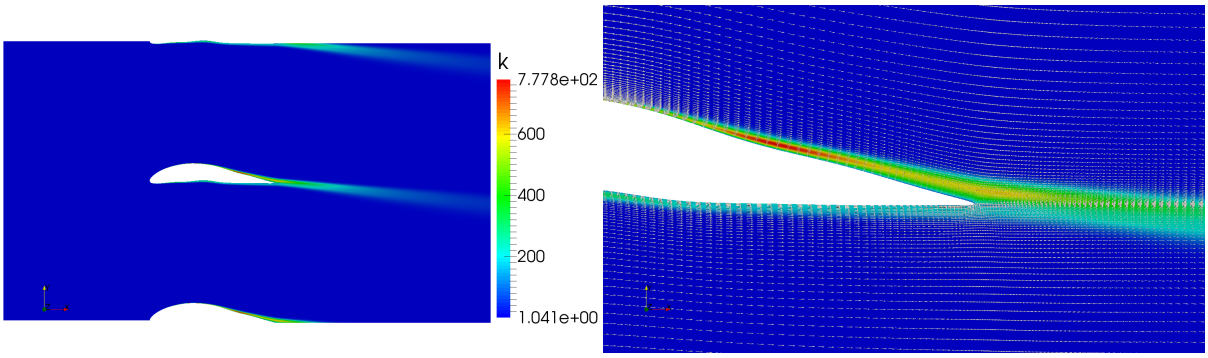
Slika 26: Polje tlaka optimirane statorske lopatice pri kutu nastrujavanja fluida od 5^o.



Slika 27: Polje Machovog broja optimirane statorske lopatice pri kutu nastrujavanja fluida od 5^o.



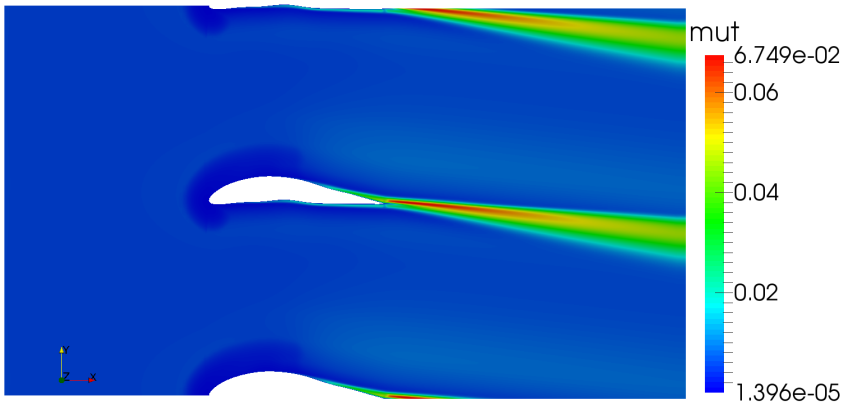
Slika 28: Temperaturno polje optimirane statorske lopatice pri kutu nastrujavanja fluida 5^o.



Slika 29: Polje turbulentne kinetičke energije optimirane geometrije statorske lopatice s detaljom izlaznog brida pri kutu nastrujavanja fluida od 5^o.

Na Sl. 29 prikazano je polje turbulentne kinetičke energije optimirane geometrije statorske lopatice s detaljom izlaznog brida pri kutu nastrujavanja fluida 5^o, dok je na Sl. 30 prikazano

polje turbulentne viskoznosti. Vidljivo je da su maksimalne vrijednosti oba polja manje u odnosu na vrijednosti dobivene za inicijalnu geometriju. Također, vidljivo je da je vrtložni trag usmjeren prema dolje, dok je kod inicijalne geometrije bio usmjeren prema gore. Na detalju izlaznog brida na Sl. 29 može se zamijetiti činjenica da bi pri daljnjem povećanju vrijednosti napadnog kuta došlo do prestrujavanja radnog fluida s pretlačne na potlačnu stranu.



Slika 30: Polje turbulentne viskoznosti optimirane geometrije rotorske lopatice pri kutu nastrujavanja fluida od 5^o.

Rješenje globalnog optimizacijskog procesa statorske lopatice je zahtjevalo 887 iteracija automatiziranog optimizacijskog procesa. MOGA je proizvela 10 generacija, što je ujedno bila i maksimalna zadana vrijednost. Usporedba rezultata inicijalne i optimirane statorske lopatice prikazana je u Tablici 5.

Table 5: Usporedba vrijednosti funkcija cilja inicijalne i optimirane geometrije statorske lopatice.

Funkcije cilja	Inicijalna geometrija	Optimirana geometrija
Δp [Pa]	2251.37	2919.64
Δu [m/s]	2.695	4.185
β	4.132^o	-6.377^o

Kut β prestavlja uprosječenu vrijednost izlaznog kuta fluida iz proračunske domene u odnosu na os x.

1 Introduction

The work presented in this thesis couples parametrisation, numerical simulation, and optimisation. First section of the thesis covers theoretical principles on which the results lay, and in the second part optimisation workflow with setup of each step in optimisation process is presented as well as the results.

1.1 An Automated Optimisation Process

Continued commercial pressure to produce turbomachinery components of highest possible aerodynamic performance within the shortest possible design cycle is one of the reasons why computational fluid dynamics is becoming a necessity. Faster design cycles, better performance and reduction in both costs and weight of the final product are just one part of the reason. Computational fluid dynamics also enables visualisation of complex flow patterns around critical components. An automated process for blade optimisation consists of the following steps:

1. Geometry parametrisation. Within this study periodic B-spline curve was used,
2. Morphing of the numerical spatial domain,
3. Numerical calculation of a compressible fluid flow through stationary turbine blade passage, and evaluation of optimisation objectives
4. Calculation of the new control polygon vertices positions using Multi-Objective Evolutionary Algorithm embedded in open source software package DAKOTA.

All numerical calculations are performed using foam-extend, a community driven fork of open-source CFD software, OpenFOAM. Optimisation was conducted for a rotor and stator blade. The same initial geometry was used for both stator and rotor, but different objective functions were prescribed. 5th degree periodic B-spline curve was used to parametrise the new blade geometry. 40 parametrisation points were extrapolated between two neighbouring control polygon vertices and the shape of a turbine blade passage was modified in accordance. By carrying out numerical calculation of compressible fluid flow in the new blade passage, objective function values were obtained. The new position of control polygon vertices was calculated using the Multi-Objective Evolutionary Algorithm and delivered back to the parametrisation step. At this point the optimisation loop restarts. The main goal of this thesis was to develop an automated turbine blade optimisation process and to investigate its robustness and time needed to obtain the most favourable result.

1.2 Thesis Outline

This thesis is divided into the following sections: Section 2, Mathematical Model, deals with partial differential equations used for conducting numerical calculation of steady compressible flow. Section 3, Blade Parametrisation, offers a brief introduction to B-spline curves, their governing properties and matrix formulation used in calculation of a closed periodic B-spline curve. In Section 4, Mesh Morphing, mesh deformation was presented while in Section 5, Multi-Objective Evolutionary Algorithm used for calculation of the new control polygon vertices. Section 6, Results of an Automated Turbine Blade Optimisation Process, deals with setup of each optimisation step presented in the optimisation workflow. Optimised results are also presented in this section. Conclusions of the thesis are given in Section 7.

2 Mathematical Model

In this chapter mathematical model for calculation of transonic flow through a turbine blade passage is presented.

2.1 Governing Equations of the Fluid Flow

Scalar transport equation in standard form is of great importance due to the fact that it models terms typical for continuum governing laws, such as convective transport, diffusive transport and volume source and/or sink terms [1]. Equation in its standard form can be expressed as follows:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) - \nabla \cdot (\gamma \nabla \phi) = q_v, \tag{1}$$

where ϕ represents the transported scalar variable, $\mathbf{u}$ represents the convective velocity and γ stands for the diffusion coefficient. The right hand side term in Eq. (1) is a source or a sink term of the transported scalar ϕ. Local volume production and destruction of ϕ is accounted for through the sources and sinks [1].

Generic transport equation consists of four characteristic terms, as follows:

- term $\frac{\partial \phi}{\partial t}$ is the temporal derivative and stands for the inertia of the system,
- term $\nabla \cdot (\mathbf{u}\phi)$ or convection term represents the convective transport of scalar ϕ by the convective velocity $\mathbf{u}$. In general terms, convective velocity can be considered as a coordinate transformation. This term has a hyperbolic nature, which by the definition means that it is defined with the direction of $\mathbf{u}$,
- term $\nabla \cdot (\gamma \nabla \phi)$ or diffusion term represents gradient transport. This is an elliptic term which means that every point (in the domain) is "aware" of every other point,
- and the term q_v. Sources and sinks (q_v) account for non - transport effects.

All of the conservation equations in the continuum mechanics are derived from the standard scalar equation. All of the governing equations will be presented as well as the simplifications used in engineering practice. Those simplifications can be divided into two groups: voluntary and involuntary. The best example of voluntary simplification is the steady state simulation, which has been used within this thesis. The involuntary ones are dictated by physical behaviour or by the flow characteristics [1]. Solution variables are: density ρ, momentum $\rho\mathbf{u}$, and energy ρe

2.1.1 Conservation of Mass

Conservation of mass or the continuity equation is derived from Eq. (1) by substituting variable ϕ with density of fluid ρ and setting source term to zero:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2}$$

In Eq. (2) the rate of change and convection represent mass transport. If mass sinks or sources would appear, they would stand on the right hand side. Notice the absence of a diffusion term due to the fact that mass does not diffuse.

2.1.2 Conservation of Linear Momentum

Conservation of linear momentum is derived from Eq. (1) by substituting ϕ with momentum (product of a fluid density and velocity field, $\rho\mathbf{u}$). Taking into account that linear momentum $\rho\mathbf{u}$ is a vector, it is possible to write conservation of linear momentum as three scalar differential equations. Again, source and sink terms are located on the right hand side, and in this equation they are equal to the sum of mass force $\rho\mathbf{g}$ and body forces $\nabla \cdot \sigma$.

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \mathbf{g} + \nabla \cdot \sigma \tag{3}$$

For compressible flow Eq. (3) can be written as:

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \nabla \cdot [\mu (\nabla \mathbf{u} + (\nabla \mathbf{u}^T))] = \rho \mathbf{g} - \nabla (P + \frac{2}{3} \mu \nabla \cdot \mathbf{u}), \tag{4}$$

where P is the dynamic pressure, μ is the dynamic viscosity and $\mathbf{g}$ is gravitational acceleration. In Eq. (3) σ is the Cauchy stress tensor. Continuity equation is coupled with the momentum equation through the rate of change of ρ because it depends on the divergence of $\rho\mathbf{u}$. Also what is important to notice is that the convective term $\nabla \cdot (\rho \mathbf{u}\mathbf{u})$ is non-linear. It's non-linearity is derived from the fact that the solution depends on itself. Thus, this term is responsible for the number of interactions in fluid flows. Presence of a non-constant density in all terms must be noted.

2.1.3 Conservation of Energy

Conservation of energy is derived from the Eq. (1). Transported scalar field ϕ is substituted with the product of fluid density and specific internal energy, ρe. Source and sink terms are

equal to the sum of mass forces $\rho\mathbf{g}\cdot\mathbf{u}$, body forces $\nabla\cdot(\sigma\cdot\mathbf{u})$ and heat flux $(-\nabla\cdot\mathbf{q}+\rho Q)$.

$$\frac{\partial\rho e}{\partial t}+\nabla\cdot(\rho e\mathbf{u})=\rho\mathbf{g}\cdot\mathbf{u}+\nabla\cdot(\sigma\cdot\mathbf{u})-\nabla\cdot\mathbf{q}+\rho Q. \tag{5}$$

In Eq. (5), e is specific energy, $\mathbf{q}$ is the heat flux and Q is a volumetric heat source. For compressible flow, Eq. (5) can be rewritten as:

$$\begin{aligned}&\frac{\partial(\rho e)}{\partial t}+\nabla\cdot(\rho e\mathbf{u})-\nabla\cdot(\lambda\nabla T)=\rho\mathbf{g}\cdot\mathbf{u}-\nabla\cdot(P\mathbf{u})-\nabla\cdot(\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{u})\\&+\nabla\cdot[\mu(\nabla\mathbf{u}+(\nabla\mathbf{u}^T))\cdot\mathbf{u}]+\rho Q.\end{aligned} \tag{6}$$

The diffusion term is given in the terms of temperature and not energy. This could pose a problem for non-constant material properties. On the right hand side there is a number of terms related to the stress tensor. The energy equation is weakly coupled with the rest of the system because e (specific internal energy), and temperature T influence ρ and convective velocity $\mathbf{u}$ through the equation of state [1]. Transport coefficient λ as well as μ are functions of variables P and T:

$$\lambda=\lambda(P,T), \tag{7}$$

$$\mu=\mu(P,T). \tag{8}$$

2.1.4 Equation of State

As stated above, to couple the energy equation with the rest of the system, an additional equation is needed. Equation of state provides the relationship between density ρ and pressure P.

$$\rho=\rho(P,T) \tag{9}$$

An ideal gas is assumed, so the equation of state can be written as:

$$\rho=\frac{P}{RT}=\psi P, \tag{10}$$

where ψ is the compressibility,

$$\psi=\frac{1}{RT}. \tag{11}$$

Compressibility effects can be observed at higher fluid velocities, and Mach number is commonly used to determine the critical velocity at which the flow can be considered

compressible. Mach number is the ratio of magnitude of fluid flow velocity and of the speed of sound. The speed of sound or velocity of weak pressure wave propagation c is calculated as:

$$c = \sqrt{\frac{\partial p}{\partial \rho}} \tag{12}$$

Physical properties of the flow depend on the local Mach number, and based on this the flow is divided into classes, as shown in Table 1

Table 1: Flow classification based on the Mach number.

low subsonic	0 - 0.3
high subsonic	0.3 - 0.6
transonic	0.6 - 1.1
supersonic	1 - 5
hypersonic	> 5

In highly subsonic flow, convective effects in the pressure distribution are significant but not dominant, while in transonic flow those effects are dominant in some parts of the flow. This implies mix of elliptic and hyperbolic nature. Also, at the moment that convective speed reaches the speed of sound shocks occurs, or in other words propagation of informations upstream is disabled. All this contributes to the fact that transonic cases are the hardest one to compute.

2.2 Coupling of Compressible Navier-Stokes Equation

Coupling of the compressible Navier-Stokes set of equations is very complex. Density appears in the momentum equation and velocity appears in the continuity equation. Energy affects density through the equation of state, and the influence of equation of state is felt both in the density and the momentum [2]. It is also important to note that compressibility effects profoundly change the nature of density-momentum coupling. This can be avoided by solving the system in an implicit block form:

$$\frac{\partial U}{\partial t} + \nabla \cdot F - \nabla \cdot V = 0 \tag{13}$$

where solution variable U is:

$$U = \begin{bmatrix} \rho \\ \rho \mathbf{u} \\ \rho e \end{bmatrix} \tag{14}$$

convective flux F:

$$F = \begin{bmatrix} \rho \mathbf{u} \\ \rho \mathbf{u}\mathbf{u} + p\mathbf{I} \\ \rho(e+p)\mathbf{u} \end{bmatrix} \tag{15}$$

and the diffusive flux V:

$$V = \begin{bmatrix} 0 \\ \sigma \\ \sigma \cdot \mathbf{u} - \mathbf{q} \end{bmatrix} \tag{16}$$

It can be seen that system is coupled across different variables as well as across the components of vectors and tensors [1].

2.3 $k-\omega$ SST Turbulence Model

When talking about turbulence models it is important to stress out that the turbulent fluid flow is governed by the Navier-Stokes equations and that the physics behind it, is fully understood. However, the problem occurs when turbulence is simulated with CFD. The fact that the numerical grid has to be fine enough to catch even the smallest vortices (the one that dissipate into heat before completing a full rotation around its axis) implies that wide span of temporal and spatial scales is needed. Therefore turbulence models come into place to provide us with physical results without the need to have very fine mesh to catch the smallest vertices (eddies). There are three modelling frameworks:

- Direct Numerical Simulation (DNS)
- Reynolds-Averaged Navier-Stokes Equations (RANS)
- Large Eddy Simulation (LES)

Within this thesis RANS model of turbulence was used. For the compressible flows with significant compressibility effects, averaging is of the Favre type. RANS model is commonly used because engineers are interested in the averaged values of e.g. forces and because turbulent flow brings significant amount of fluctuation. Favre averaging will account for density weighting. Let Φ be any dependant variable which is decomposed into mean and fluctuating part, as presented in Eq. (17)

$$\Phi = \widetilde{\Phi} + \Phi'', \tag{17}$$

where

$$\widetilde{\Phi} = \frac{\overline{\rho\Phi}}{\overline{\rho}}. \tag{18}$$

In Eq. (18) overbars denote averaging using Reynolds decomposition [8]. Using Favre averaging on Navier-Stokes equations, compressible turbulent flow governing equations are obtained [2].

The $k-\omega$ SST (Shear Stress Transport) [2] turbulence model is the two equation model where k stands for turbulent kinetic energy and it is defined by:

$$\frac{\partial k}{\partial t} + \nabla\cdot(\mathbf{u}k) - \nabla\cdot[(\nu+\sigma_k\nu_T)\nabla k] = P_k - \beta^* k\omega) \tag{19}$$

and ω is eddy turn-over time and is defined as:

$$\frac{\partial \omega}{\partial t} + \nabla\cdot(\mathbf{u}\omega) - \nabla\cdot[(\nu+\sigma_\omega\nu_T)\nabla\omega] = \alpha S^2 - \beta\omega^2 + 2(1-F_1)\sigma_{\omega,2}\frac{1}{\omega}\nabla k\cdot\nabla\omega \tag{20}$$

Turbulent viscosity is calculated as:

$$\nu_T = \frac{\alpha_1 k}{max(\alpha_1\omega, SF_2)}, \tag{21}$$

where closure coefficients and auxiliary relations are as follows:

$$P_k = min(\boldsymbol{\tau}\mathbf{:}\nabla\mathbf{u}, 10\beta^* k\omega), \tag{22}$$

$$F_2 = tanh\left[\left[max\left(\frac{2\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega}\right)\right]^2\right], \tag{23}$$

$$F_1 = tanh\left\{\left\{min(\left[max\left(\frac{\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega}\right), \frac{4\sigma_{\omega 2}k}{CD_{k\omega}y^2}\right]\right\}^4\right\}, \tag{24}$$

and

$$CD_{k\omega} = max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\nabla k\mathbf{:}\nabla\omega, 10^{-10}\right) \tag{25}$$

$k-\omega$ SST model was developed by Menter in 1993 and in combines the best properties from $k-\omega$ model by Wilcox and the $k-\varepsilon$ model. Wilcox's $k-\omega$ model is used from the inner parts of the boundary layer down to the wall (through viscous sub-layer). SST formulation switches to the $k-\varepsilon$ model behaviour in the free-stream region [8]. By melding these two models together, it is possible to use this turbulence model without near-wall damping functions. Constants in the differential equations of $k-\omega$ *SST* are tuned so that the

model would imitate the actual physical behaviour of turbulent fluid flow as best as possible.

2.4 Mathematical and Physical Boundary Conditions

The role of boundary conditions is to isolate space which is being investigated from its surroundings. Appropriate choice of boundary conditions requires sufficient knowledge of the observed problem. For example, placing an outlet boundary condition in recirculation zone is an ill defined system. [1].

Numerical boundary conditions are:

(a) Dirichlet boundary condition, or condition in which transported scalar value ϕ is given on the boundary and it is fixed,

(b) Neumann boundary condition or zero gradient (no flux) boundary condition. Analytically it can be expressed as $\mathbf{n} \cdot \mathbf{q}_s = 0$. Additional generalisation of Neumann condition is possible and that is fixed gradient or fixed flux boundary condition. Or analytically: $\mathbf{n} \cdot \mathbf{q}_s = q_b$,

(c) Mixed boundary condition which is a linear combination of Dirichlet and Neumann boundary conditions (mixed boundary condition = α Dirichlet + $(1 - \alpha)$ Neumann).

Additional classification of numerical boundary conditions is possible and it regards reduction in mesh size (cell count) in order to decrease computational demands. Those boundary conditions are:

(a) Symmetry plane
This boundary condition can be used if these requirements have been satisfied:

- the geometry is symmetric,
- boundary conditions are symmetric,
- the flow is steady,
- expected flow pattern is symmetric

If all these conditions are met, it is allowed to model only a section of the problem.

(b) Cyclic and periodic boundary conditions
These conditions can be used in cases of repeating geometry, for example stator blades or rotor blades. Only representative segment of the geometry has to be calculated [1].

When dealing with compressible flow, it is important to keep in mind that three variables $(\rho, \mathbf{u}, e)$ have to be handled together. In supersonic cases, at the inlet all three variables are specified while at the outlet no variables are specified. When talking about subsonic or transonic cases, at the inlet two variables are specified and at the outlet one variable is specified [1].

2.5 Closure

In this chapter mathematical model used for fluid flow calculation was presented. In the next chapter blade parametrisation will be presented.

3 Blade parametrisation

In the previous chapter mathematical model used to describe compressible fluid flow was presented while in this chapter blade parametrisation will be discussed.
The blade optimisation process is an iterative one and the first step is parametrisation of the turbine blade geometry. The geometry is represented with a set of points which are then used to morph the initial computational mesh. B-spline was used for blade parametrisation.

3.1 B-spline Curve Definition

B-spline basis (from Basis Spline) is generally non-global, and its non-global nature is a result of the fact that each vertex B_i is associated with a unique basis function [3]. As a result, each vertex affects the shape of a function only over a range of parameter values where support function is non-zero. B-splines theory was first suggested by Schoenberg [11] and the recursive definition was independently discovered by Cox [9] and by de Boor [10]. If $P(t)$ is the position vector along a curve and a function of a variable t, B-spline curve can be expressed analytically as [3]:

$$P(t) = \sum_{n=1}^{n+1} B_i N_{i,k}(t) \qquad t_{min} \leq t < t_{max}, \qquad 2 \leq k \leq n+1, \tag{26}$$

where $N_{i,k}(t)$ are normalized basis functions, and B_i are position vectors of $n+1$ control vertices. Generally looking for ith normalized basis function of order k, basis functions $N_{i,k}(t)$ are defined by the Cox-de Boor recursion formulas:

$$N_{i,1}(t) = \begin{cases} 1 & \text{if} \quad x_i \leq t < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

where

$$N_{i,k}(t) = \frac{(t - x_i) N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t) N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}. \tag{28}$$

Values of x_i have to be a series of monotonically increasing real numbers where the relation $x_i \leq x_{i+1}$ is satisfied. This series of increasing real numbers is called a knot vector. Due to the fact that Cox-de Boor formula is recursive, a basis function of order k depends on the basis functions of lower orders down to the order 1 [3].

3.1.1 Properties of B-spline curves

B-spline curve is defined as a polynomial spline function of order k that satisfies the following conditions:

1. Position vector $P(t)$ along the curve is a polynomial of order k (degree $k-1$) for every interval $x_i \leq t < x_{i+1}$
2. $P(t)$ and all of its derivatives are continuous over the entire curve,

and due to the fact that a B-spline basis are used to describe a B-spline curve, the following properties can be derived:

1. For every value of parameter t, the sum of B-spline functions has to satisfy the relation:

$$\sum_{n=1}^{n+1} N_{i,k}(t) \equiv 1, \tag{29}$$

2. $N_{i,k} \geq 0$ (value of any basis function has to be positive or zero for every value of parameter t),
3. All basis functions have precisely one maximum value (except for the value of order $k = 1$ because that implies that the degree of the curve is 0)
4. The curve can not be of larger order than the number of its control vertices,
5. The curve both follows the shape, and lies within the convex hull of its control polygon,
6. The shape of the curve is changed by changing the positions of its control polygon vertices while sustaining its variation-diminishing property [3].

3.1.2 Knot vectors

From Eq. (27) it can be seen that the choice of the knot vector has a great influence on the B-spline basis function and thus on the B-spline curve. When talking about knot vectors it is important to note that it is a monotonically increasing series of real numbers (as stated before) which are generally divided into periodic and open ones. Each type of a knot vector can be further divided into uniform and nonuniform ones. Uniform knot vectors have evenly spaced individual knot values, as given in example below:

$$[0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8],$$

or normalized as:

$$[0 \quad 0.125 \quad 0.25 \quad 0.375 \quad 0.5 \quad 0.625 \quad 0.75 \quad 0.875 \quad 1].$$

Periodic uniform basis function is defined using periodic uniform knot vector of the order k (degree $k-1$) as follows:

$$N_{i,k}(t) = N_{i-1,k}(t-1) = N_{i+1,k}(t+1) \tag{30}$$

which means that every basis function is in fact a translation of another one. An open uniform knot vector is the one that can be described as follows:

$$x_i = 0 \qquad 1 \leq i \leq k$$

$$x_i = i - k \qquad k+1 \leq i \leq n+1$$

$$x_i = n - k + 2 \qquad n+2 \leq i \leq n+k+1$$

or simply the one that has a number of the same knot values at the end, and at the beginning of a knot vector. An example of an open uniform knot vector can be seen below [3]:

$$k = 2 \qquad [0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 4],$$

or in normalized form:

$$k = 2 \qquad [0 \quad 0 \quad 0.25 \quad 0.5 \quad 0.75 \quad 1 \quad 1].$$

For nonuniform knot vectors there are no mathematical expressions for generating a knot vector, neither for nonuniform open knot vector nor for nonuniform periodic knot vector. The reason for this lies in the fact that nonuniform knot vectors can have unequally spaced and/or multiple knot values [3]. Examples of nonuniform knot vectors can be seen below as follows:

1. Open nonuniform knot vector:

$$[0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2],$$

2. Periodic nonuniform knot vector:

$$[0 \quad 0.2 \quad 0.48 \quad 0.64 \quad 0.86 \quad 1].$$

3.1.3 B-spline Curve Control

There are a number of control types with which the resulting shape of a B-spline curve can be influenced. Due to the flexibility of B-spline basis function, a number of control types and control type combinations are possible. Control of a B-spline curve is achieved by [3]:

1. Changing the type of a knot vector,
2. Changing the order k (or degree $k-1$) of the basis function,
3. Removing or adding control vertices
4. Changing the position of control vertices
5. Using a number of identical control vertices
6. Using multiple knot values in the knot vector

3.2 Periodic B-spline Curves

Periodic B-spline curves are generated using periodic knot vectors, and they are very useful for generating closed curves. For closed periodic B-spline curves it is important to note that $k-2$ control vertices have to be repeated on the beginning or/and at the end, otherwise the resulting B-spline curve would not be closed.

3.2.1 Matrix Formulation of Periodic B-spline Curves

Important properties of periodic B-spline curves are as follows: influence of a basis function is limited to k intervals, and B-spline basis functions are all translation of each other. Therefore, all basis functions on the interval $0 \leq t^* < 1$ have the same form, $N^*_{i,k}(t^*)$ [3]. By reparametrisating the basis functions to this interval, a point on a parametric B-spline curve is defined as follows:

$$P_j(t^*) = \sum_{i=0}^{k-1} N^*_{i+1,k}(t^*) B_{j+i} \qquad 1 \leq j \leq n-k+1, \quad 0 \leq t^* < 1 \tag{31}$$

where j is a curve segment counter and n is one less than the number of control vertices. Written in matrix form it stands:

$$P_j(t^*) = [T^*][N^*][G] = [F][G] \qquad 1 \leq j \leq n-k+1, \qquad 0 \leq t^* < 1 \tag{32}$$

or in other words the blending function matrix $[F]$ multiplied with a geometry matrix $[G]$. Geometry matrix $[G]$ contains control vertices :

$$[G] = \begin{bmatrix} B_j \\ . \\ . \\ . \\ B_{j+k-1}, \end{bmatrix} \tag{33}$$

and $[F]$ is composed of B-spline basis functions, and can be written as:

$$[F] = [T^*][N^*] \tag{34}$$

where $[T^*]$ is

$$[T^*] = [t^{*k-1} \quad t^{*k-2} \quad \cdots \quad t^* \quad 1] \qquad 0 \leq t^* < 1 \tag{35}$$

and Cohen and Riesenfeld [12] presented a generalized form of $[N]$ as:

$$[N^*] = [N^*_{i+1,j+1}] \tag{36}$$

where $N^*_{i+1,j+1}$ is:

$$N^*_{i+1,j+1} = \frac{1}{(k-1)!}\binom{k-1}{i}\sum_{l=j}^{k-1}(k-(l+1))^i(-1)^{l-j}\binom{k}{l-j} \qquad 0 \leq i,j \leq k-1 \tag{37}$$

3.3 Parametrisation of a Turbine Blade Profile

B-spline parametrisation is used to obtain points which belong to the turbine blade profile. As mentioned in Section 3.1.3, B-spline was used because of its favourable properties: it approximates the control vertices and does not match them exactly, and it is easily controlled by changing the position of the control polygon vertices. In this way, discontinuities on the blade surface are avoided. Each segment of the resulting periodic curve was of the 6th order (5th degree) and 40 parametrisation points were extracted between each pair of control vertices. On Fig 1 parametrised NACA 4412 airfoil can be seen. The red Xs represent control

vertices and black dots denote points of parametrisation.

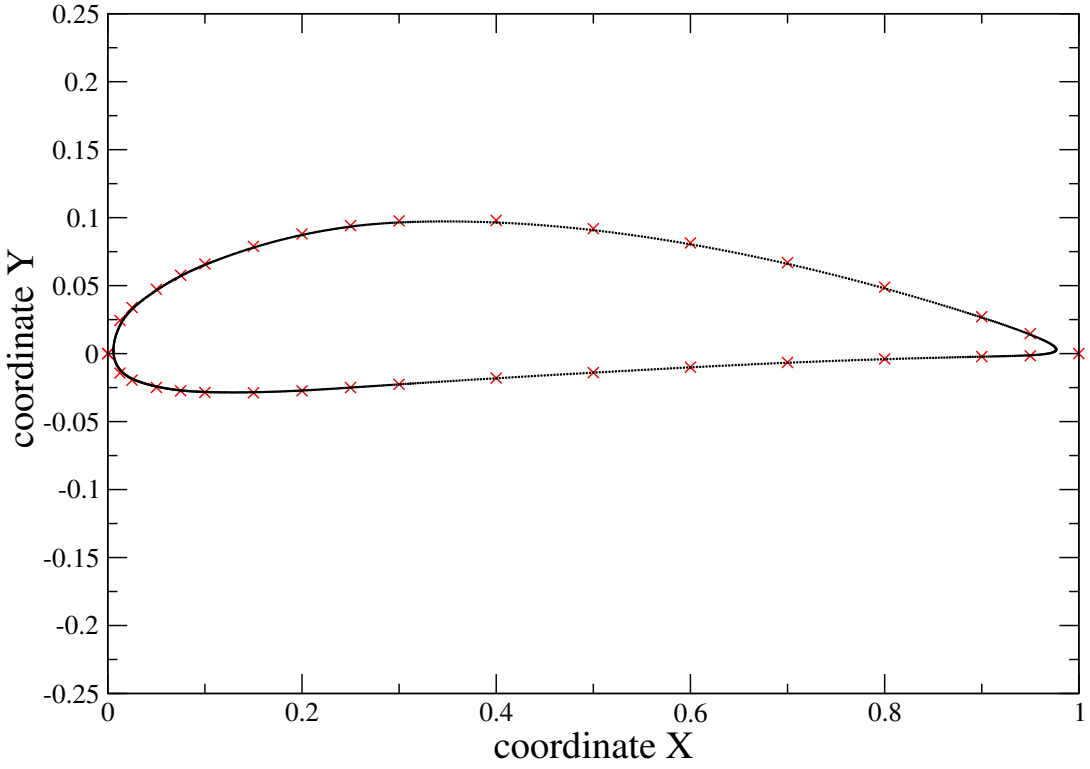
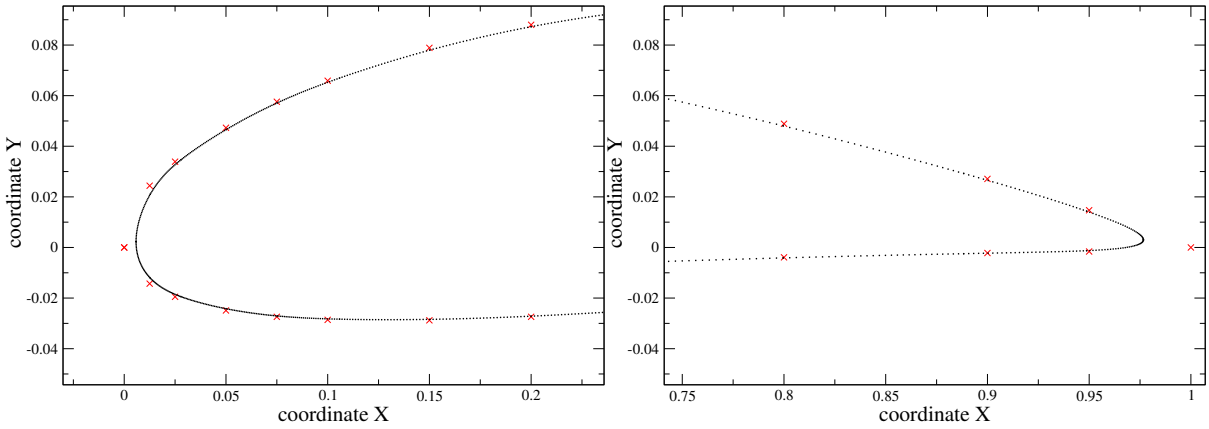


Figure 1: NACA 4412.

In Fig 2 details of the leading and trailing edge of NACA 4412 airfoil are shown.



(a) NACA 4412 Leading edge

(b) NACA 4412 Trailing edge

Figure 2: NACA 4412 Leading and trailing edge.

3.4 Closure

In this chapter definition of a B-spline curve was given. Main properties of B-spline basis functions, and knot vectors were discussed. Matrix formulation of implemented periodic B-spline curve was presented. In the next chapter mesh morphing will be presented.

4 Mesh Morphing

In the previous chapter blade parametrisation using a B-spline curve, knot vectors, and the matrix formulation of a periodic B-spline curves were presented. In this chapter morphing of a numerical mesh will be discussed.

In [14] an overview of numerical errors resulting from the discretisation procedure (treating convection and temporal terms) and mesh-induced errors is given. Consistent use of discretisation schemes in blade passage calculations can be ensured, but mesh-induced errors are more difficult to measure and control. A large number of geometries are tested by the optimisation algorithm and it is time demanding to create a new computational mesh for each geometry. In order to compare the results for different geometries, similar mesh resolution and quality (orthogonality, skewness) must be ensured for each case. Creating the computational mesh by hand for every single geometry to have more control over the resulting mesh properties, is impossible. Automatic meshing algorithms cannot provide the same or even similar mesh configuration for different geometries, no matter how small the shape variation. This is the reason a mesh morphing procedure was developed for blade geometry optimisation.

The morphing procedure is based on assumption that the computational mesh behaves as an elastic body in a state of equilibrium, as described in [15]. If some force acts on the boundary of the body, it will cause the motion of the boundary which will influence the interior of the body. Influence of boundary motion on interior points of the mesh can be described with a Laplace equation:

$$\nabla \cdot (\gamma \nabla \mathbf{u}) = 0 \tag{38}$$

where $\mathbf{u}$ is the displacement vector, and γ is the diffusion coefficient. By applying the Laplace displacement equation with a constant coefficient γ, the largest deformation of the mesh elements occurs near the mesh boundary. It usually leads to extreme mesh distortion and negative cell volumes. This is why a space dependent coefficient of diffusivity is used. Notice that increase of the diffusivity coefficient results in lower deformation of the mesh element. Thus, it is beneficial to define the diffusivity coefficient to be inversely proportional to distance of the mesh element from the mesh boundary. The function can be linear, quadratic or exponential [15]. In this study, a quadratic function is used:

$$\gamma(l) = l^{-2}, \tag{39}$$

where l is the minimum distance of the mesh element from the mesh boundary.

The Laplace displacement equation Eq. (38) is discretised into a linear system of equations:

$$[A] \cdot \mathbf{u}_k = r_k, \quad k = 1,3, \tag{40}$$

where **A** is the coefficient matrix, **u** is the unknown vector of displacement and **r** is the right hand side vector. The new geometry configuration is defined with a set of points which are introduced into the system Eq. (40) as a fixed value (Dirichlet) boundary condition.

4.1 Closure

In this section morphing of a numerical mesh was presented. In the next chapter properties of the optimisation algorithm implemented in the open source software DAKOTA will be described.

5 Optimisation

In the previous chapter, a mesh morpher algorithm was described. After obtaining the results of the numerical calculation, they are processed by the optimisation algorithm implemented in the open source software DAKOTA [6]. In this chapter, the properties of the optimisation algorithm will be described.

5.1 Optimisation Capabilities

Optimisation algorithm is used to minimize and/or maximize an objective function(s) related to the performance of the engineering component under consideration. In a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis, DAKOTA, both gradient-based and derivative-free local, and global methods are available, as well as more advanced algorithms such as surrogate-based minimization and multi-objective optimisation [6].

5.1.1 Mathematical Formulation of an Optimisation Problem

In this section a brief introduction to the mathematical formulation of optimisation will be presented. A general optimisation problem is defined as:

$$\begin{aligned} \textit{minimize}: \quad & f(\mathbf{x}) \\ & \mathbf{x} \in \Re^n \\ \textit{subject to}: \quad & \mathbf{g}_L \leq \mathbf{g}(\mathbf{x}) \leq \mathbf{g}_U \\ & \mathbf{h}(\mathbf{x}) = \mathbf{h}_t \\ & \mathbf{a}_L \leq \mathbf{A}_i\mathbf{x} \leq \mathbf{a}_U \\ & \mathbf{A}_e\mathbf{x} = \mathbf{a}_t \\ & \mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \end{aligned} \tag{41}$$

Bold capital letters are matrix terms, and bold small letters are vector terms. $\mathbf{x}$ stands for an n-dimensional vector of real valued design variables, or mathematically $\mathbf{x} = [x_1, x_2, \ldots, x_n]$. Design variables or design parameters have n-dimensional vectors $\mathbf{x}_L$ and $\mathbf{x}_U$ as the lower and upper bound of the design parameters, respectively. The set of all values that lie between these bounds form a design or parameter space. Therefore, any set of values in design space are dubbed as design points [6].

As stated before, the goal is to minimize or maximize an objective function. For a general optimisation problem in Eq. (41), the optimisation goal is to minimize an objective function $f(x)$ while satisfying the constraints. Those constraints are categorised as follows:

- Nonlinear inequality constraints, **g(x)** which have lower and upper bounds ($\mathbf{g}_L$, $\mathbf{g}_U$ respectively),
- Nonlinear equality constraints, **h(x)** which have target values, or $\mathbf{h}_t$
- Linear inequality constraints create a linear system $\mathbf{A}_i\mathbf{x}$ in which $\mathbf{A}_i$ is a coefficient matrix of the system, and $\mathbf{a}_L$, and $\mathbf{a}_U$ are the lower and upper bounds respectively.
- Linear equality constraints create a linear system $\mathbf{A}_e\mathbf{x}$ in which $\mathbf{A}_e$ is the coefficient matrix of the system, and $\mathbf{a}_t$ are the target values.

Constraints partition the design space in a feasible region, or the region of parametric space for which all of the constraints are satisfied and infeasible regions in which one or more constraints are violated. Many different methods for solving optimisation problems given in Eq. (41) exist, and all of them iterate on **x** in one fashion or other. An initial value for each parameter in an n-dimensional vector of real valued design variables, **x** is chosen and the response quantities $f(\mathbf{x})$, **h(x)**, **g(x)** are calculated [4].

5.2 Evolutionary Algorithm

Evolutionary algorithm (EA) is inspired by Darwin's theory of evolution. By using mechanisms inspired by biological evolution: reproduction, mutation, and selection an optimal result is obtained. In other words through number of trial and mistakes the best possible solution appears. EA is divided into several steps, where each step represents one basic principle of evolutionary theory [4]:

1. The initial population of individuals (first generation) is randomly generated
2. Fitness of each individual in population is evaluated
3. The best-fit individuals for reproduction are determined
4. New individuals through crossover, survival and averaging are obtained
5. Mutation on the offspring is performed
6. Fitness of new individuals is determined and survival of the fittest is implemented

Within this thesis genetic algorithm, the most popular type of EA was used. Genetic algorithm is a derivative free global method. Derivative free methods are usually more robust than the gradient based approaches and they are applied when the problem is non-smooth, multi modal or poorly behaved. On the other hand this approach is usually more computationally demanding because of its slower convergence rates for finding an optimal

solution(s). Multi-objective Genetic Algorithm, MOGA, an algorithm specially designed for multi-objective problems [4] is implemented in developed optimisation process.

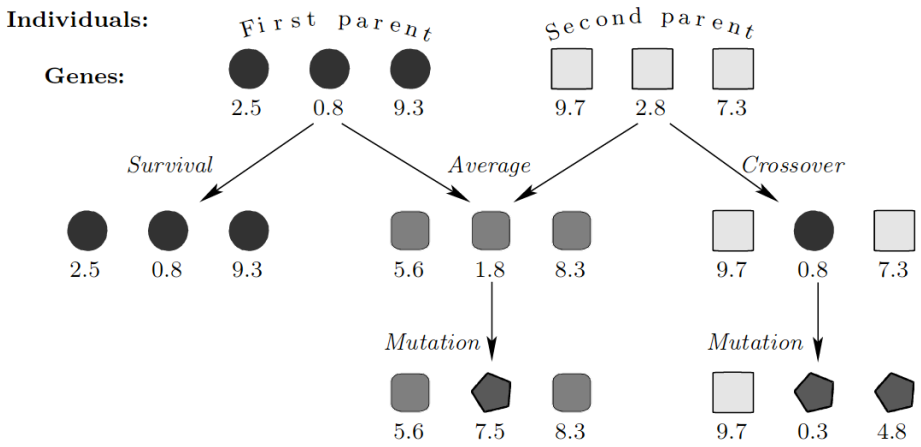


Figure 3: Basic principle behind EA [5].

5.2.1 Multi-Objective Genetic Algorithm

MOGA is a global optimisation method that performs Pareto optimisation and it is a part of JEGA library which is embedded in DAKOTA. This algorithm is best suited for problems involving a multi-modal space, but the possibility that the process will result in suboptimal solution still exists [4]. The basic steps of the algorithm are:

1. Population initialization

   The initial population is randomly generated, and each individual in the population has a set of design point values that form a "genetic string". Initialization can be performed as follows:

   - with duplicates allowed (one or more identical individuals are allowed).
   - warm-start approach

   It is encouraged not to allow duplicates, if it isn't necessary. In that way design point revaluation is avoided and design space exploration is encouraged. As a result, both wasting memory and suboptimal convergence are avoided.

2. Initial population members evaluation

   For each member from the initial population, objective functions are calculated. For each member from the initial population constraints are calculated and used to find the fittest "individuals". Values of objective functions can be calculated either by using external software or by using analytical functions embedded in DAKOTA.

3. Perform crossover

   From initial pool of individuals, mating pairs are chosen. Each mating pair can produce a (user input) number of offspring (later on referred as children).

4. Perform mutation

   Mutation happens on randomly chosen children, and the rate of mutation is a user defined variable.

5. New population member evaluation

6. Fitness assessment of each member in the population

   In MOGA, convention says that higher fitness represents better result.

7. Insert children into population

   In this step the Pareto dominance of the design is considered. After creating children and performing mutation on them, procedure of inserting them into population begins.

   - Infeasible members exist in the population
     The child with the highest fitness is determined and chosen for comparison. If the child demonstrates higher fitness then the fitness of compared counterpart, its entrance in the population is allowed. Otherwise, entrance in the population is prohibited.
   - All members in the population are feasible
     Again the child with the highest fitness is determined and chosen for comparison. Algorithm keeps track of the "weakest" member in the population and that one is chosen for comparison with the child. If the fitness of a child has a greater value than the fitness of the weakest member in the population, entrance of the child in the population is allowed. If the child does not demonstrate higher fitness than any member of the population, its entrance in the population is prohibited.

8. Choose members for reproduction
   Creation of a mating pool based on the design fitness values is performed. In a mating pool all members of the population are present, but the ones with the higher design fitness have a higher chance of producing offspring. That is achieved by increasing the number of individuals with more favourable design fitness, i.e. they appear more than once in a mating pool. Mating pairs are created by randomly choosing individuals from the mating pool. The number of individuals that will be selected is determined by [4]:

   (a) the size of a population,

(b) the number of children per crossover,

(c) the user specified crossover rate.

This iterative process is performed until one of the stopping criteria is met:

(a) number of iterations is equal to maximum number of iterations

(b) number of function evaluations is equal to the maximum number of function evaluations

or the optimization process has produced the best possible result by defined inputs, i.e. percentage of change of the results is negligible.

5.2.2 Pareto Optimality

In multi-objective optimisation it is necessary to resolve two or more (usually conflicting) objective functions. Results obtained from optimisation present a set of possible results that are equivalent in quality, or simply said if there are two or more conflicting objective functions no single optimum can be found. That is why "Pareto optimality" or "Pareto dominance" concept is used. In Pareto dominance, design A dominates design B if and only if [4]:

$$(\forall i f_i(A) \geq f_i(B)) \cap (\exists j : f_j(A) > f_j(B)), \tag{42}$$

for which f_i is the i-th objective function in process of maximization of n objectives. Eq. (42) reads: at least one aspect design A is better than design B, while in all others they can be equal [4]. Now, when dealing with two objective functions f_1, and f_2, which need to be optimised, for each evaluation [f_1, f_2] couple is produced. All couples produced, or all data produced, represent a design set called Pareto front or Pareto optimal set, whose dimension is equal to $n-1$ [4].

5.3 Closure

In this chapter Evolutionary algorithm, and MOGA were discussed. Basic mathematical formulation of an optimisation problem and pareto optimality were presented. In the next chapter results of an automated optimisation process will be discussed.

6 Results of an Automated Turbine Blade Optimisation Process

In the previous chapter optimisation using Evolutionary Algorithm was presented and Multi-Objective Genetic Algorithm was explained in detail. In this chapter an overview of the workflow setup and results will be given.

6.1 Problem Description

A two-dimensional model of turbine blade passage was considered. Optimisation was performed both for the rotor and stator. Evaluation of the stator blade geometry was performed by setting the maximal increase of the fluid velocity (Δu) and the minimal pressure drop (Δp) as objective functions. Two values of attack angles were examined. The initial computational domain used in optimisation of both stator and rotor blade is shown in Fig 4. Objective functions, used for rotor blade optimisation were: maximal value of lift and drag coefficient ratio ε, and minimal value of drag coefficient c_D.

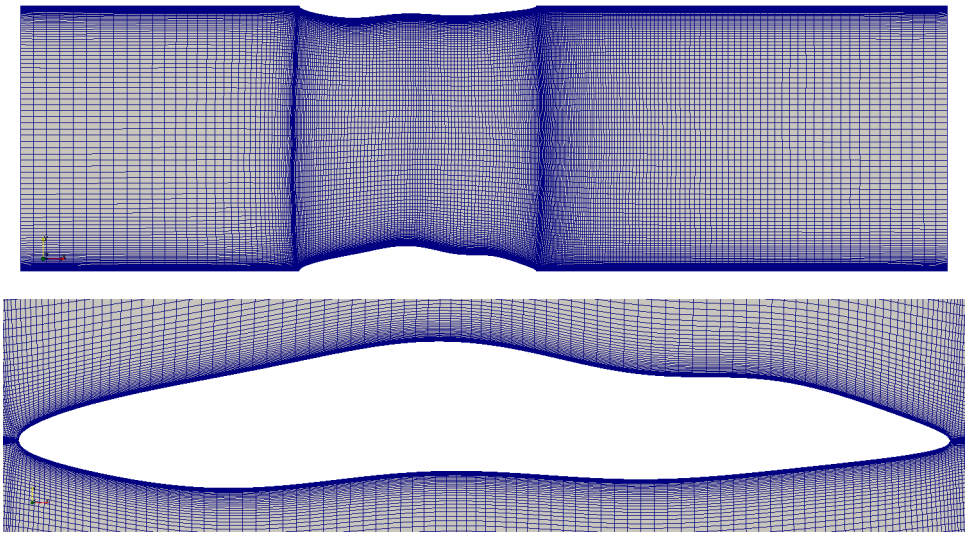


Figure 4: Initial numerical spatial domain and the boundary layer.

For almost all cases, transonic flow conditions were achieved in the channel. The outer dimensions of the domain (2.5 x 1 m) as well as the position of the two control vertices were constant. The two control vertices define the position of the leading and trailing edge. By keeping their position fixed, equal length of blade chords were assured. Positions of 8 control vertices on the upper side and 8 control vertices on the lower side of the blade were manipulated only in tangential direction (in respect to y axis).

6.2 Numerical Spatial Domain

Numerical spatial domain is fully structured and it was created using commercial mesh generating software *Pointwise*. This mesh generating software is non-automatic which enables complete control on the mesh resolution and quality.

6.2.1 Mesh Refinement Study

Two meshes were compared, one with approximately 28 000 cells, Fig 5, and one with approximately 75 000 cells, Fig 6. Steady state compressible flow through turbine blade passage was simulated. NACA 4412 blade profile was used for mesh refinement study and the results were compared.

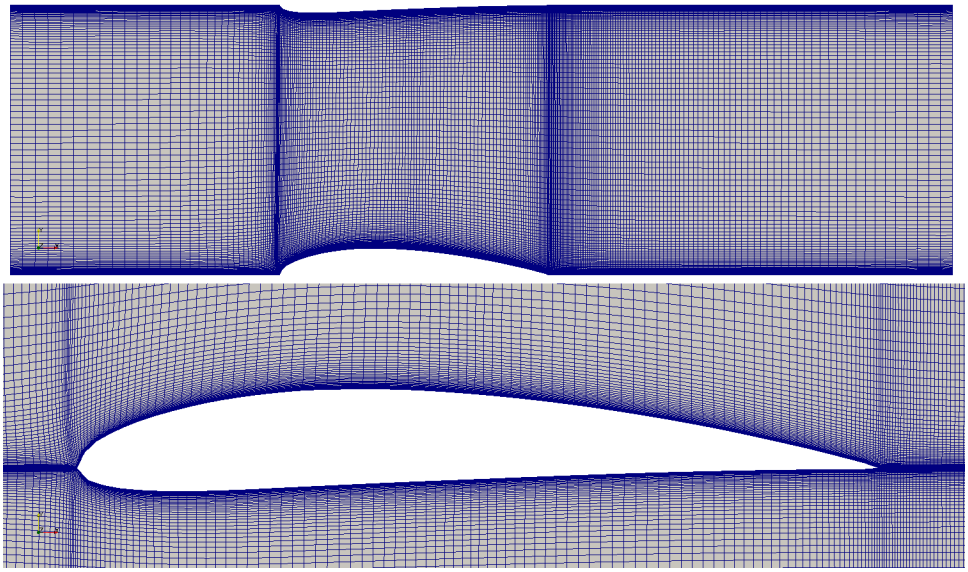


Figure 5: Coarse mesh: 28 000 cells.

Fine mesh has a thicker boundary layer and thus, more precise results are to be expected. In Table 2, mesh quality measures are shown for fine and coarse mesh.

Table 2: Properties of coarse and fine meshes.

Mesh	A	B
Number of cells	28203	75603
Max aspect ratio	403.40536	403.40536
Max non-orthogonality	49.055129	54.161589
Max skewness	1.5335585	1.5335585

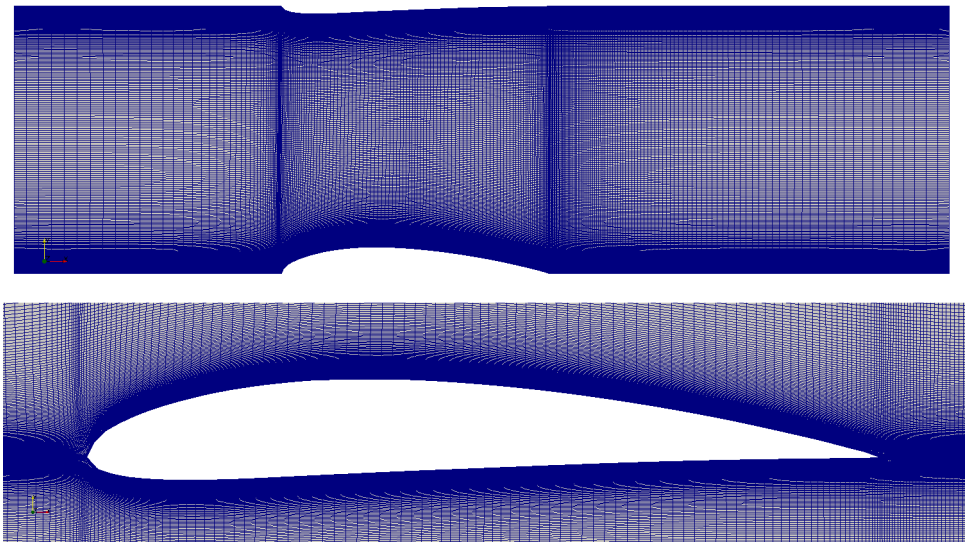


Figure 6: Fine mesh: 75 000 cells.

NACA 4412 airfoil is often used as a rotor blade profile. Thus, in the mesh refinement study values of drag coefficient c_D and drag and lift ratio coefficient ε were calculated. In Table 3 values of calculated lift and drag coefficients for the attack angle $\alpha = 0^o$ are shown.

Table 3: Objective function values of coarse and fine meshes.

Mesh	Coarse	Fine
Lift coefficient c_L	0.30055247	0.25739102
Drag coefficient c_D	0.054778846	0.042831108
lift drag coefficient ratio ε	5.49	6.01

It was decided to use the coarse mesh due to a large number of numerical calculations which had to be conducted and limited CPU power. Also fine mesh demonstrated oscillatory convergence for some cases. The fact that results of courser and finer mesh slightly differ, does not diminish the capabilities of the optimisation process due to the fact that the same initial mesh is used. The best possible global solution will be discovered but further investigation of exact obtained values have to be conducted through a mesh refinement study.

6.3 Software Coupling and Optimisation Workflow

In this section software packages used in the automated blade optimisation process of a stationary turbine blade passage will be further discussed. The process is divided into the following steps which describe the geometry with most favourable properties:

1. Parametrisation

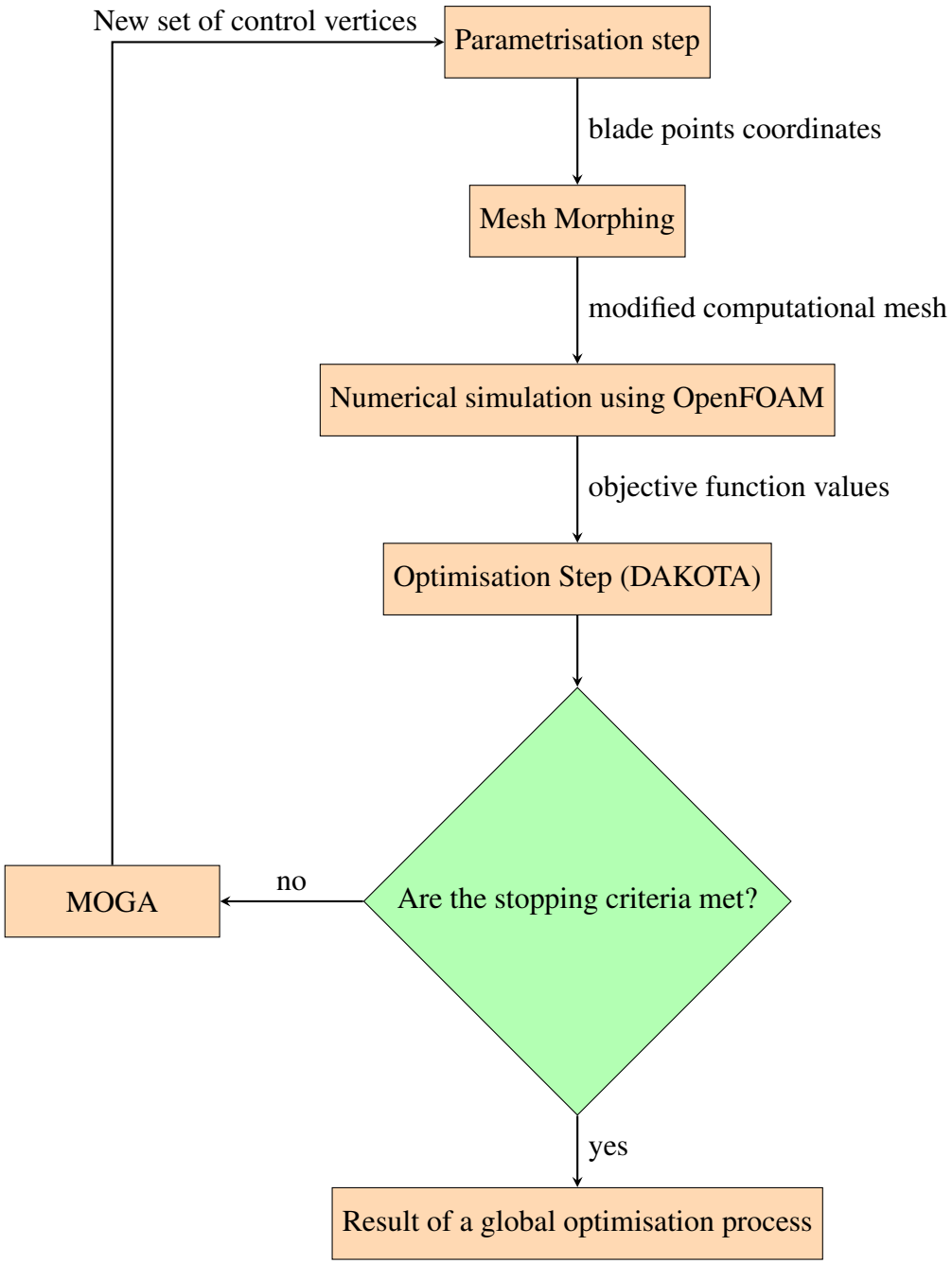
   Location of the control polygon vertices is used to fit an approximation curve, and parametrisation points are obtained and written in a file.

2. Mesh Morphing
   Using the initial mesh, shown in Fig. 4, a new modified numerical spatial domain is obtained by applying parametrisation points.

3. CFD Simulation
   Steady compressible fluid flow through the turbine blade passage is calculated and values of objective functions are obtained.

4. Optimisation step
   Values of objective functions from OpenFOAM simulation are delivered to DAKOTA optimisation software and new values of control vertices positions are proposed by the evolutionary algorithm and delivered back to the first step.

The details of the workflow with description of communication between workflow components are given in Appendix A. Steps two and three are performed using OpenFOAM. In it tools for mesh manipulation, `meshMorpher` utility and range of features for calculation of physical problems are included. In this thesis `steadyCompressibleFoam` was used to calculate the compressible steady state airflow through the stationary turbine blade passage. By solving the set equations presented in Chapter 2, values of objective functions were obtained. Those values were delivered to DAKOTA optimisation software and, using MOGA, new positions of control polygon vertices were calculated and written in a file. After completion of step four, a python script was used for calculation of the parametrisation points according to the algorithm described in Chapter 3. This iterative procedure was repeated until the termination criteria were met.



6.4 Parametrisation Setup

Eight control vertices on the upper side and eight control vertices on the lower side of the blade are all equidistant. Control polygon vertices are described with the curve of 5th degree, and between each two neighbouring control polygon vertices 40 parametrisation points are extrapolated. B-spline curve control, or the resulting shape of a turbine blade geometry, is obtained by changing the position of control vertices in tangential direction in reference to the blade section. As stated in Chapter 3, by implementing B-spline curve any discontinuities along the turbine blade profile were avoided. Parametrisation was performed using a periodic

B-spline curve. Degree of the curve was chosen in such fashion that a sufficient number of control vertices would influence each segment of the curve as suggested in [13].

6.5 OpenFOAM Setup

As stated before open source CFD software OpenFOAM is used for mesh manipulation and numerical simulation of compressible airflow through turbine blade passage.

6.5.1 Computational Mesh Description

After successful completion of mesh refinement study, coarser mesh was chosen. The domain of this 2D mesh consists of 28 203 cells, and it was created as suggested in the best practice guidelines for turbomachinery CFD [8]. Flow-path simulations are usually performed on structured hexahedral meshes and the neighbouring cell size should not be larger than a factor of 1.25. Along the suction and the pressure side of a blade it is recommended to have 100 cells in streamwise direction, and the boundary layer should have from 10 to 20 cells. Leading and trailing edges have to be resolved very carefully to avoid and discontinuity. The size of a first cell, as well as the cell growth ratio, in the boundary layer for a high Reynolds number, should yield value of y^+ in range between 20 and 200 [8]. Recommended growth ratio factors are in range 1.1 - 1.2 [8]. Coarser mesh was created with respect to all conditions, so it has 10 cells in the boundary layer, 100 cells in streamwise direction on the pressure and on the suction side, and the growth ratio is of factor 1.1. Additionally, great care was paid to the refinement on the leading and trailing edge, as it is presented in Fig 7.

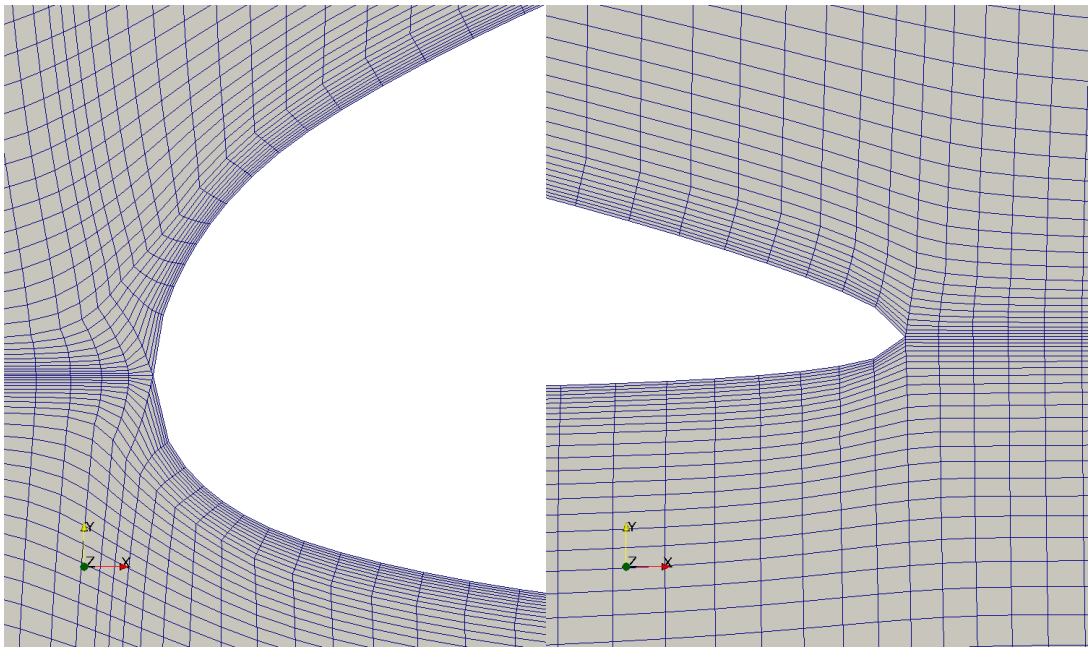


Figure 7: Boundary layer on the leading and trailing edge of coarser mesh.

Value of the y^+ in all cases run during the optimisation process was in range 50 ± 5. Initial (background) mesh used in the optimisation process, both for the stator and for the rotor blade was created by upholding all of the stated criteria, and its boundary layer can be observed in Fig 8, while the initial turbine blade passage mesh is shown in Fig 4.

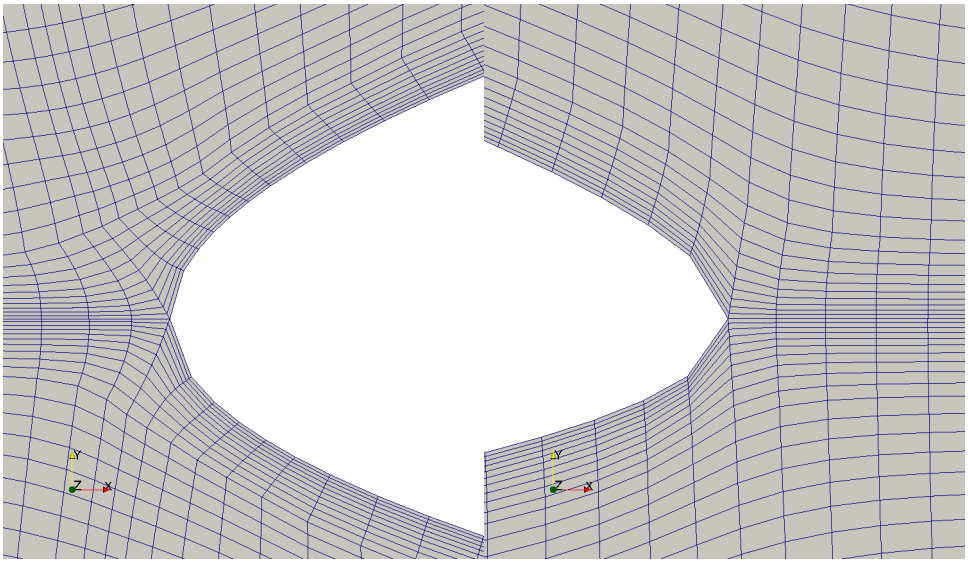


Figure 8: Leading and trailing edge boundary layer on the initial (background) mesh.

6.5.2 Mesh Morphing

In the second step of an automated optimisation process morphing of the numerical mesh is performed. This is done with `meshMorpher` utility within OpenFOAM.

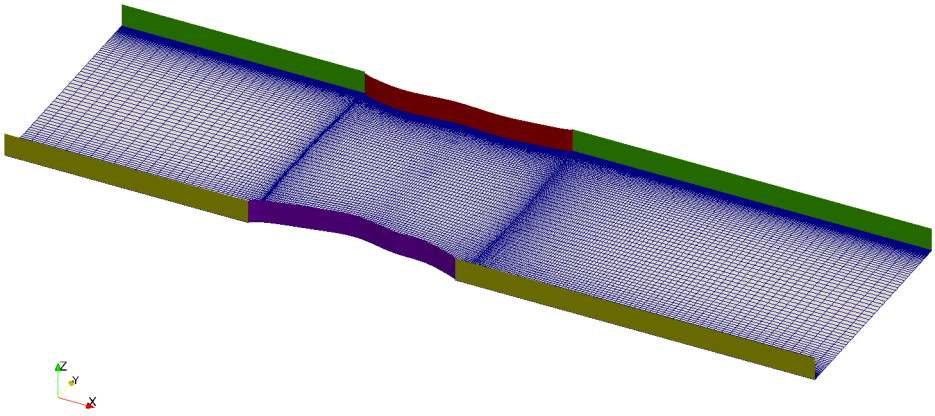


Figure 9: Initial mesh with detail of periodic and shadow patch.

In Fig 9, the pressure (red) and suction side (purple) are shown. These patches are used in

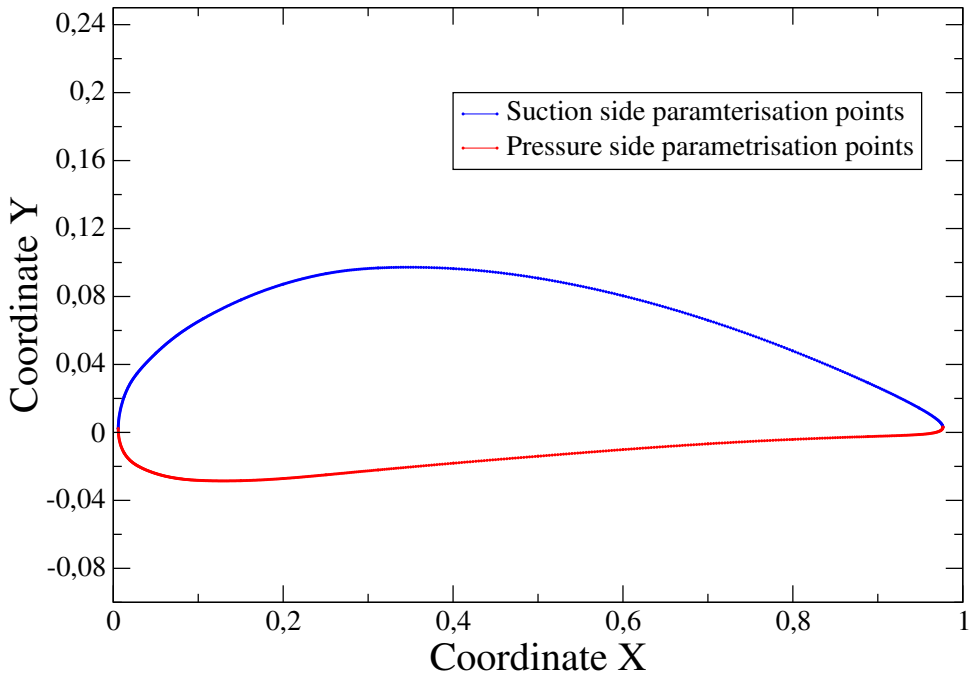


Figure 10: Parametrisation points describing the pressure and suction side of a blade.

the morphing process as the initial position of points coordinates. A new list of points is given by the parametrisation algorithm and used as the boundary condition for the mesh morphing algorithm. Coordinates describing new position of the blade suction and the pressure side are shown in Fig 10.

Morphing of the mesh is performed in two iterations. Initial turbine blade passage pressure and suction side are shown in Fig 11

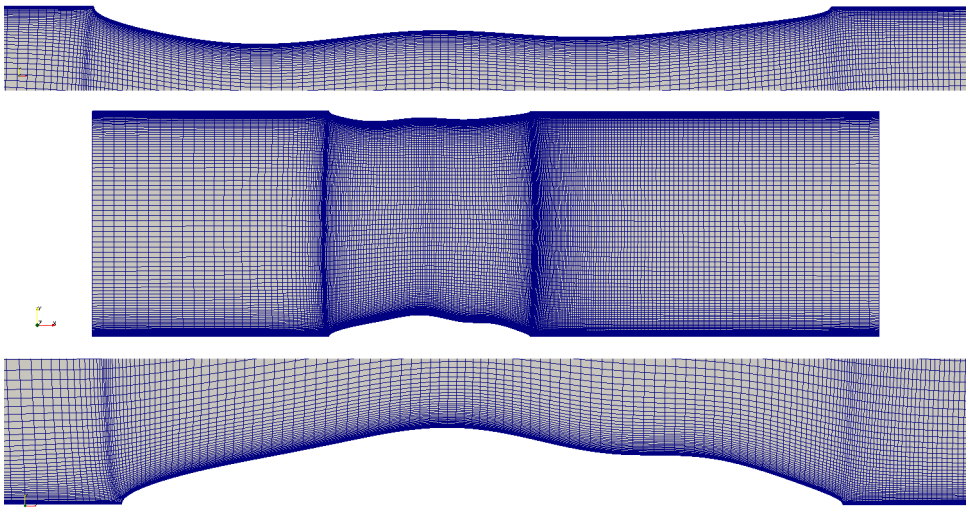


Figure 11: Initial geometry of a turbine blade passage with detailed pressure and suction side of the blade.

In the first iteration, mesh is modified in the axial direction so that the points of the pressure

and suction side match the ones given as the boundary condition. Boundary conditions for the first iteration of mesh morpher are shown in Table 4. In Fig 12, turbine blade passage after the first iteration of the morphing is shown.

Table 4: Boundary conditions used for mesh deforming in axial direction.

Patch	Boundary condition
Inlet	slip
Outlet	slip
blade_ps	Dirichlet
periodic	slip
blade_ss	Dirichlet
shadow	slip
Front	empty
Back	empty

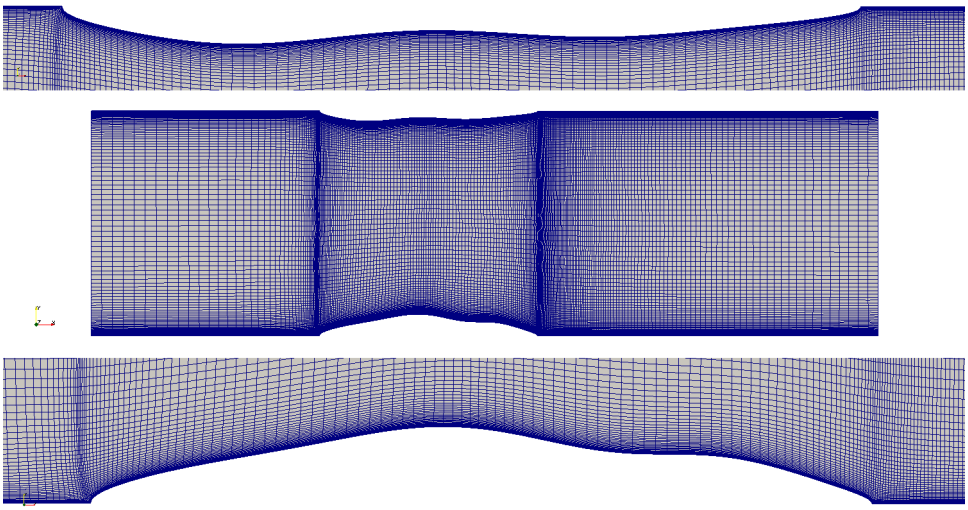


Figure 12: Geometry of the turbine blade passage after the first iteration of the morphing.

In the second iteration, mesh deformation is performed in the tangential direction in respect to the blade section. Boundary conditions used in this step are shown in Tab 5.

Table 5: Boundary conditions used for mesh deforming in tangential direction.

Patch	Boundary condition
Inlet	slip
Outlet	slip
blade_ps	Dirichlet
periodic	Dirichlet
blade_ss	Dirichlet
shadow	Dirichlet
Front	empty
Back	empty

Table 5, shows that in the second iteration patches `periodic`, and `shadow`, as well as, `blade_ss`, and `blade_ps`, which correspond to pressure and suction side, have a fixed value boundary condition. This was done to preserve the channel width. Theoretically, the channel could expand or contract to conform to blade thickness. The final mesh obtained after the second iteration of mesh morphing is shown in Fig 13

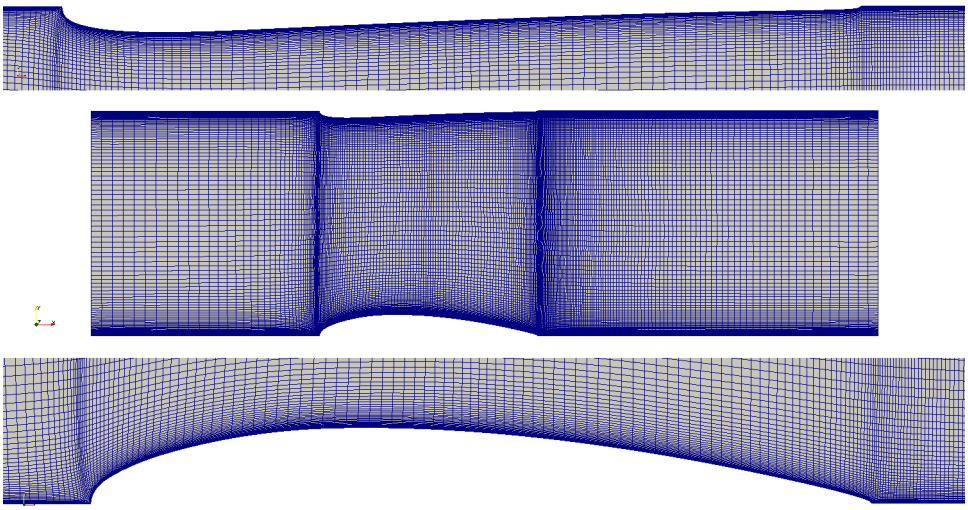


Figure 13: Final numerical mesh obtained with detailed pressure and suction side of a blade.

Morphing of the mesh is run using `meshMorpher` command. After the mesh morphing stage was performed, file `Points` created in the second iteration is copied into the numerical simulation step.

6.5.3 Numerical Simulation

Turbulence parameters were calculated using next empirical expressions

- Turbulent kinetic energy k:

$$k = \frac{3}{2}(uI)^2, \tag{43}$$

- Specific dissipation ω
  - For rotor blade:

$$\omega = 600k, \tag{44}$$

  - For stator blade:

$$\omega = 300k, \tag{45}$$

where u is the free stream velocity and I is the turbulent intensity which was for rotor blade optimisation set to 0.5%, and for stator blade optimisation set to 0.4% [8].

Boundary conditions (BC) used for numerical calculation are summarized in Table 6.

Table 6: Boundary conditions used in a numerical simulation of a compressible flow.

Patch	**Velocity BC**	**Pressure BC**	**Temperature BC**
Inlet	`pressureDirectedInletVelocity`	`isentropicTotalPressure`	`isentropicTotalTemperature`
Outlet	`inletOutlet`	Dirichlet	Neumann
blade_ps	Dirichlet	Neumann	Neumann
periodic	`cyclicGGI`	`cyclicGGI`	`cyclicGGI`
blade_ss	Dirichlet	Neumann	Neumann
shadow	`cyclicGGI`	`cyclicGGI`	`cyclicGGI`
Front	empty	empty	empty
Back	empty	empty	empty

Generalised grid interface [16] is used for coupling of interfaces of non-conformal regions which means that patch nodes of the two coupled regions do not match. Direct interpolation is present in the GGI where the flow values are interpolated directly from one interface patch to another. Even though the initial mesh is completely conformal to ensure the optimisation process running smoothly any possible deviation of patch nodes had to be avoided. In cases, where coupling interfaces are pairs of cyclic GGIs, flow values are evaluated and transmitted across a pair of non-conformal patches using weighted interpolation [16]. Patches for which GGI was used are shown in Fig 9, coloured green (`periodic`) and yellow (`shadow`).
In the optimisation process of a rotor blade, static pressure value at the outlet was set to 100 000 Pa for Mach number value 0.6, and for calculation of total values of temperature and pressure Eq. (46), and Eq. (47) were used:

$$\frac{p_{tot}}{p_{stat}} = \left(1 + \frac{\kappa - 1}{2} M^2\right)^{\frac{\kappa}{\kappa - 1}}, \tag{46}$$

$$\frac{T_{tot}}{T_{stat}} = 1 + \frac{\kappa - 1}{2} M^2, \tag{47}$$

where p_{tot} is the total pressure, p_{stat} is the static pressure, T_{tot} is the total temperature, T_{stat}

is the static temperature, and M is the Mach number and which is calculated as:

$$M = \frac{u}{\sqrt{\kappa RT}}, \tag{48}$$

where R is the individual gas constant, and κ the adiabatic index is defined as:

$$\kappa = \frac{c_p}{c_v}. \tag{49}$$

In Eq. (49) c_p is the heat capacity at constant pressure, and c_v is the heat capacity at constant volume. Static temperature at the outlet was set to 288.15 K.

In the optimisation process of a stator blade static pressure on the outlet was set to 150 000 Pa, and values of total pressure and total temperature were calculated for value of the Mach number $M = 0.6$ using Eq. (46), and Eq. (47) respectively. Used static temperature was 350 K. `pressureDirectedInletVelocity` was used as a inlet velocity boundary condition because it allows the definition of attack angle of the fluid flow. Value of velocity on the blades was set to 0 while temperature and pressure were calculated using the Neumann boundary condition. As described in Section 2.4, numerical calculation is defined with two variables at the inlet, in this case `isentopicTotalTemperature` and `isentropicTotalPressure`, and one variable at the outlet (in this case fixed value of static pressure).

6.6 DAKOTA Setup

After the numerical simulation was conducted, objective function values were delivered to the optimisation step, where by using MOGA embedded in DAKOTA, new positions of control vertices were calculated. Objective functions are physical values derived from flow variables, which describe the characteristics of each blade geometry. Multi-objective optimisation is run using `dakota.in` file which can be found in Appendix B, and Appendix C for the rotor, and for the stator blade optimisation, respectively.

In this section, setup of the MOGA algorithm will be presented. Maximal number of function evaluation equaled 10 000 even tough this value was never reached. Thus, in all run cases maximal number of iterations was considered as user defined stopping criterion. Sixteen variables, or sixteen values of control polygon vertices were bounded according to Fig 14 and Fig 15. .

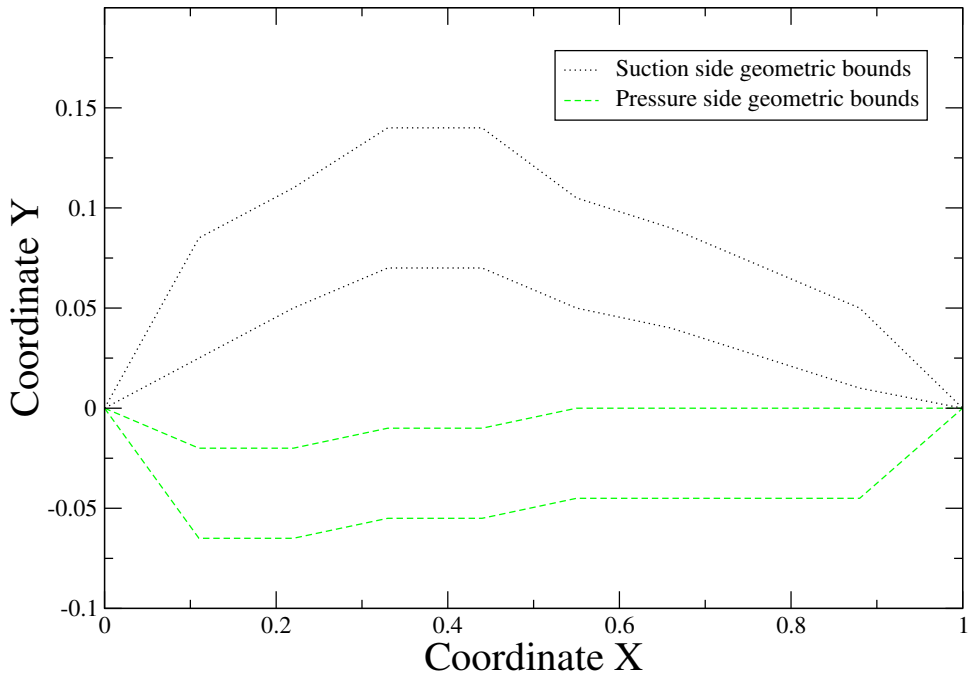


Figure 14: Geometric bounds for a rotor turbine blade.

Geometric bounds, or continuous bounding of examined variables is of great importance. These values must be specified very carefully because e.g., a flat plate with appropriate attack angle can have lift to drag coefficient ratio of 10. Thus, suboptimal convergence due to bad geometric bounding is a possibility.

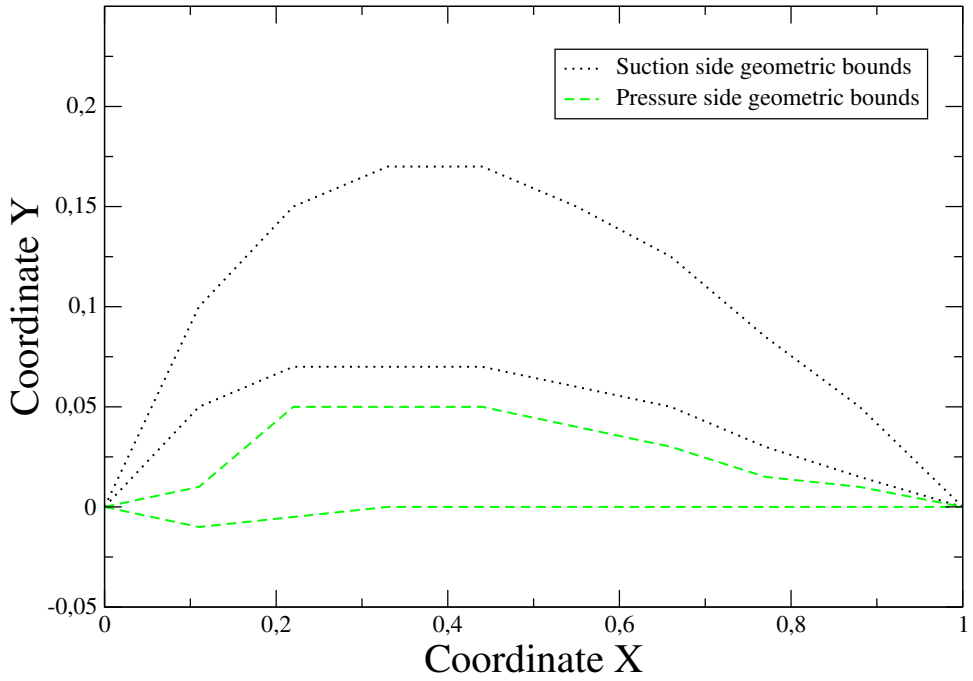


Figure 15: Geometric bounds for a stator turbine blade.

Key options used to run MOGA are presented in Table 7 as suggested in [4], and detail

description of all options can be found in [7].

Table 7: MOGA options setup.

`initialization_type`	`unique_random`
`crossover_type`	`shuffle_random`
`mutation_type`	`offset_normal`
`fitness_type`	`domination_count`
`replacement_type`	`below_limit`
`niching_type`	`max_designs`

6.7 Results of Rotor Blade Optimisation

In turbines power is produced by expanding the fluid to the lower pressure, and in doing so dynamic action of a rotating element, rotor, is employed. In other words, rotor extracts energy from the fluid flow and converts it, as much as possible, to useful mechanical work. Thus, objective functions for rotor blade optimisation were: lift drag coefficient ratio ε, and the drag coefficient c_D, with sense max and min respectively. The drag coefficient c_D is calculated as:

$$c_D = \frac{F_D}{\frac{1}{2}\rho u^2 A}, \tag{50}$$

where F_D is the drag force, and A is the relevant plan area calculated as the product of the blade span and the blade chord.

$$c_L = \frac{F_L}{\frac{1}{2}\rho u^2 A}; \tag{51}$$

where F_L is the lift force, and the ε is defined as:

$$\varepsilon = \frac{c_L}{c_D} \tag{52}$$

Optimisation process was tested for rotor blade at two different values of attack angle; 2.5^o, and 7.5^o. In all cases starting geometry was the same.

6.7.1 Attack Angle $\alpha = 2.5^o$

In this section, a comparison of initial blade geometry and the resulting geometry of the optimisation process will be given. Irregular and distorted starting geometry of turbine blade was used to demonstrate robustness of developed optimisation process.

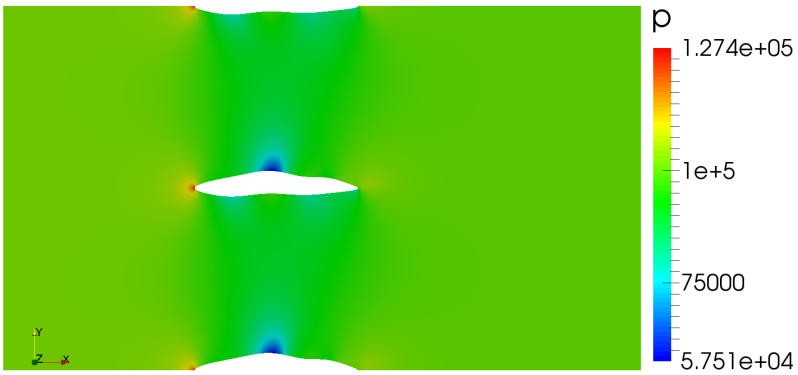


Figure 16: Pressure field for the initial blade geometry for the attack angle $\alpha = 2.5^o$.

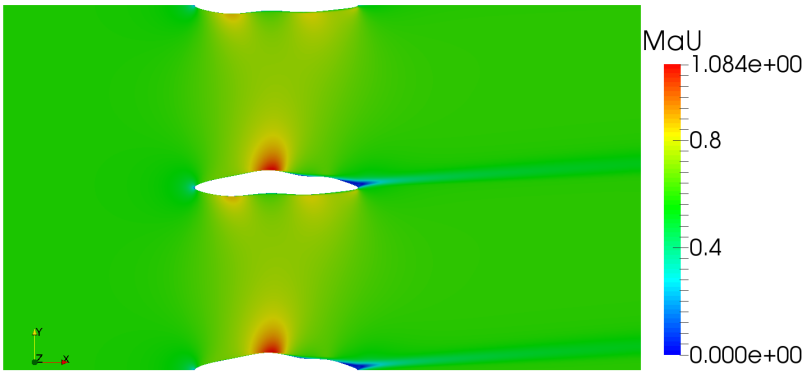


Figure 17: Mach number field for the initial blade geometry for the attack angle $\alpha = 2.5^o$.

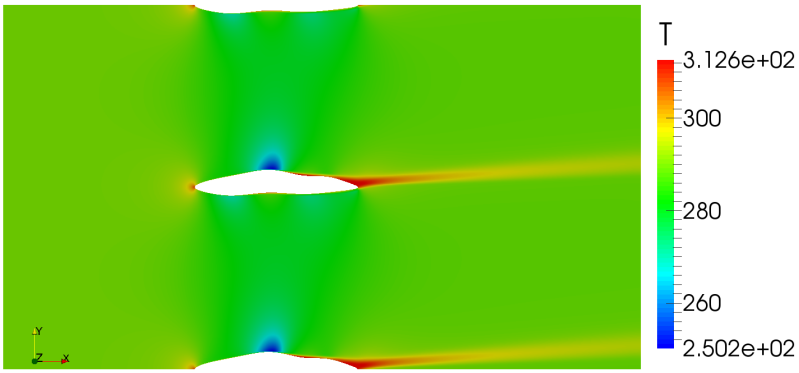


Figure 18: Temperature field for the initial blade geometry for the attack angle $\alpha = 2.5^o$.

In Fig 16, Fig 17, and Fig 18 pressure field, Mach number field and temperature field of compressible flow through initial turbine blade passage can be observed, respectively. It can be seen that due to the incoherent shape of the initial geometry, number of localised pressure and temperature drops on both pressure and the suction side of the blade exist. In the same locations, due to the Bernoulli principle, increase in fluid velocity can be observed. Furthermore, in Fig 17, and Fig 18 the direction of the fluid exiting the blade passage can be determined, and from Fig 19, and Fig 20 the wake forming behind the initial blade geometry can be observed. In Fig 17 it can be observed that fluid flow through turbine blade passage is a fully transonic.

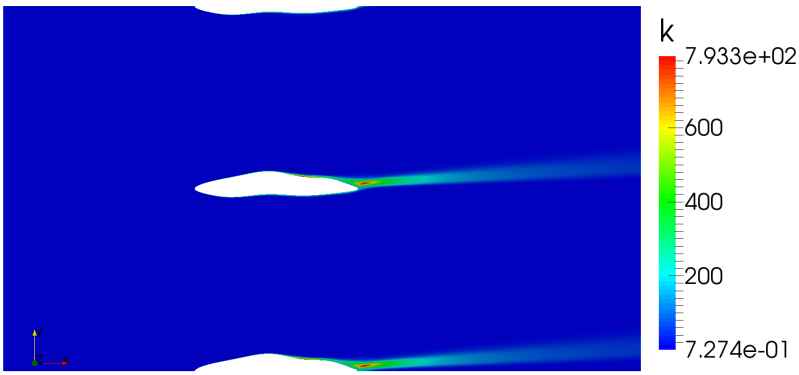


Figure 19: Turbulent kinetic energy field for the initial blade geometry for the attack angle $\alpha = 2.5^o$.

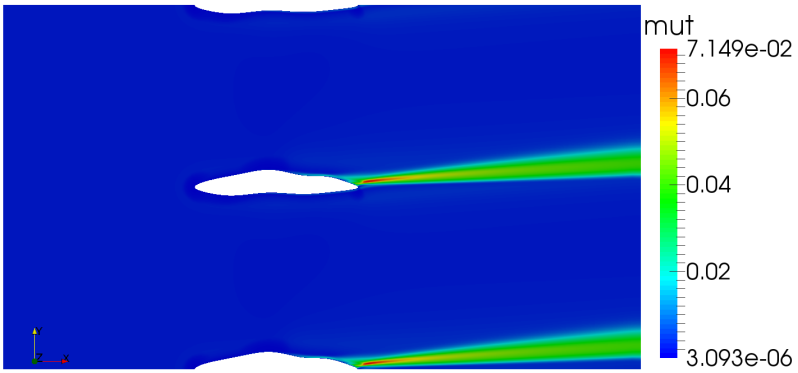


Figure 20: Turbulent viscosity field for the initial blade geometry for the attack angle $\alpha = 2.5^o$.

In Fig 19, and Fig 20 turbulent kinetic energy field and turbulent viscosity field can be observed, respectively. The wake forming near the trailing edge of the blade can be observed as well as the fact that it is directed upwards. The values of objective functions are calculated as follows:

(a) $\varepsilon = -1.0822$

(b) $c_D = 0.0677$

Due to the incoherent shape of the initial geometry, relatively high maximum values of both turbulent kinetic energy and the turbulent viscosity field are obtained. Negative value of ε means that this blade geometry generates negative lift for 2.5^o attack angle.

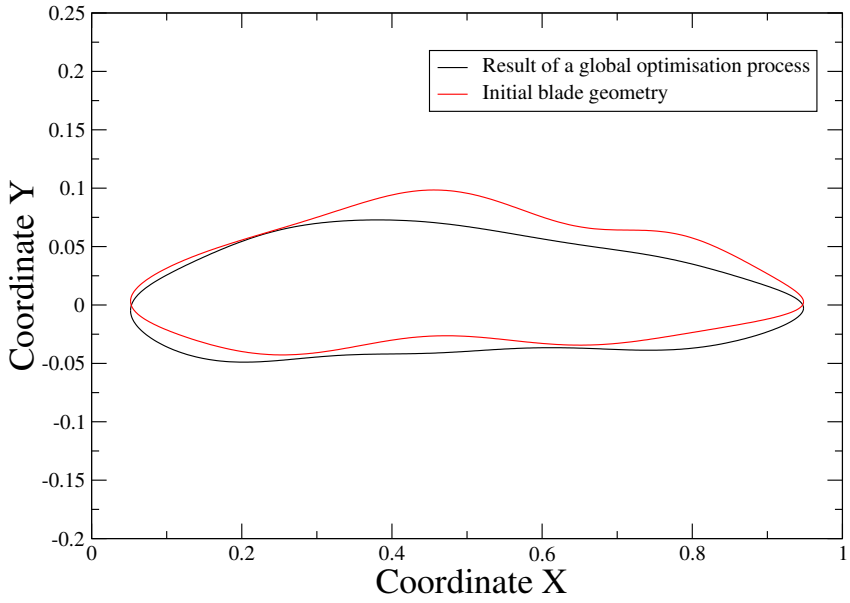


Figure 21: Comparison of the initial and optimised blade geometry for attack angle $\alpha = 2.5^o$.

In Fig 21, a comparison of the initial and optimised blade geometry for 2.5^o attack angle is shown. In Fig 22 optimised blade geometry within its geometric bounds is shown. Also, the distribution of the control polygon vertices along the blade suction and pressure side can be observed. The fact that B-spline curve only approximates and does not interpolate control polygon vertices assures no discontinuity along the blade surface, as it can be seen in Fig 22.

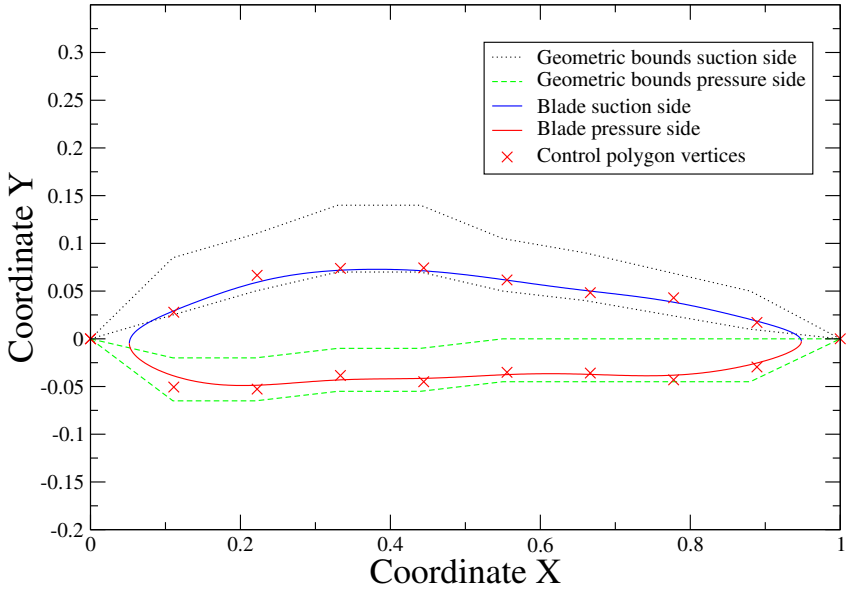


Figure 22: Optimal blade geometry solution of a global optimisation process for the attack angle $\alpha = 2.5^o$.

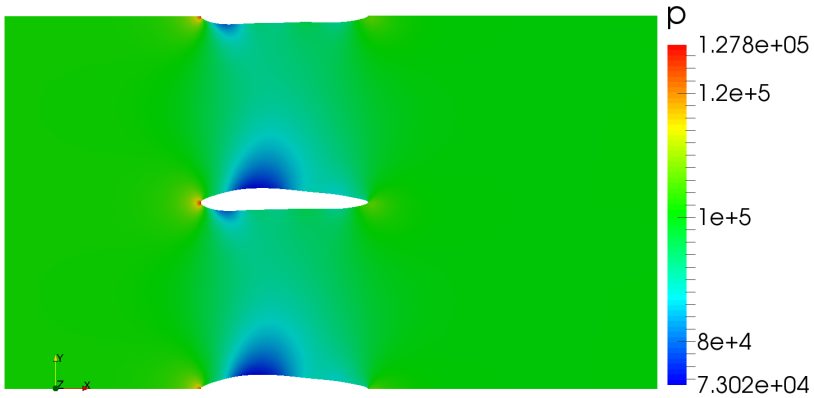


Figure 23: Pressure field for the optimised rotor blade geometry for the attack angle $\alpha = 2.5^o$.

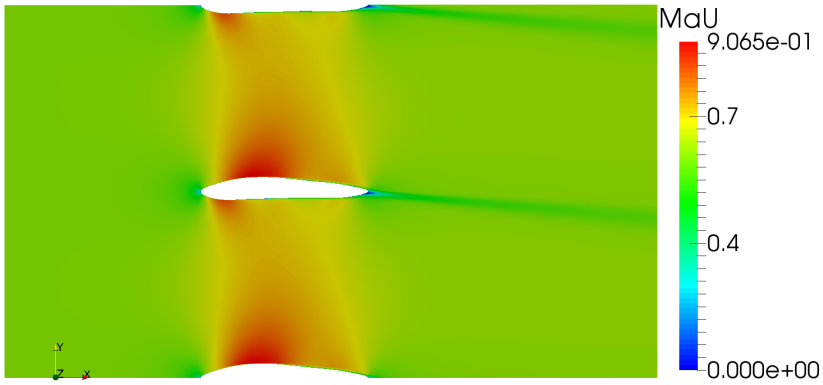


Figure 24: Mach number field for the optimised rotor blade geometry for the attack angle $\alpha = 2.5^o$.

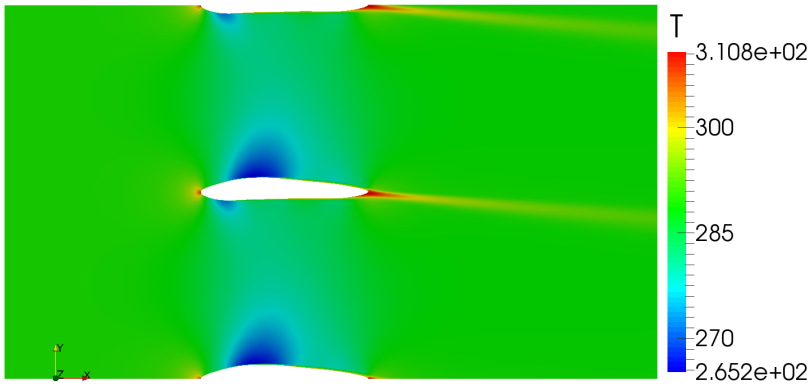


Figure 25: Temperature field for the optimised rotor blade geometry for the attack angle $\alpha = 2.5^o$.

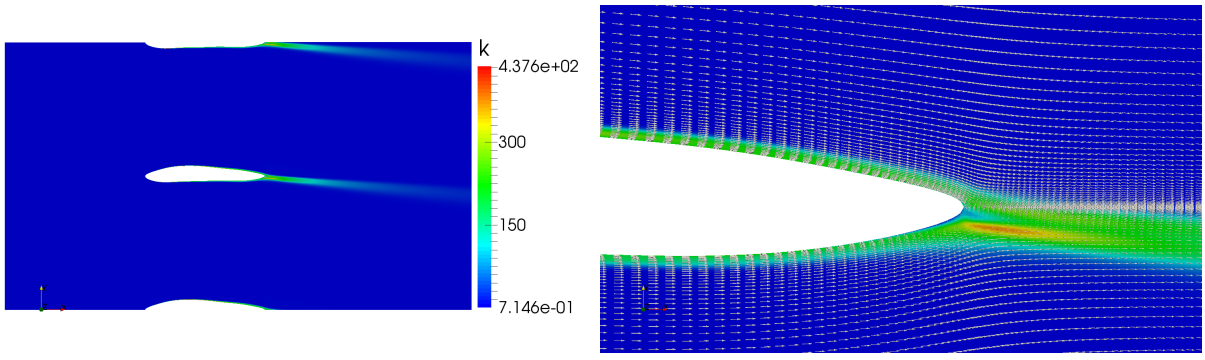


Figure 26: Turbulent kinetic energy field with trailing edge detail for the optimised rotor blade geometry for the attack angle $\alpha = 2.5^o$.

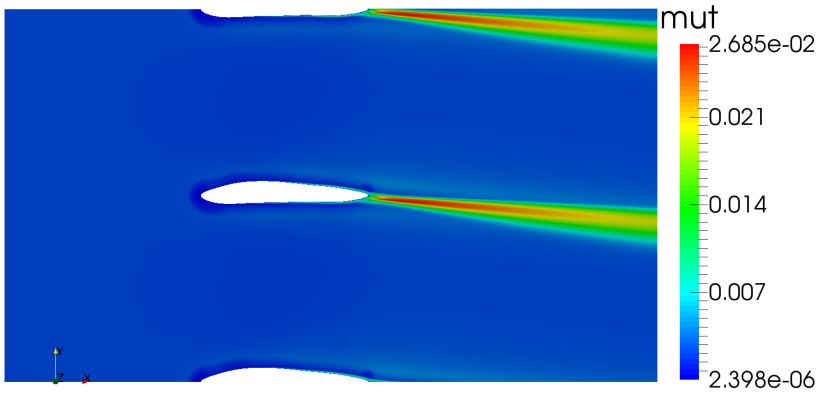


Figure 27: Turbulent viscosity field for the optimised rotor blade geometry for the attack angle $\alpha = 2.5^o$.

In Fig 23, Fig 24, and Fig 25 pressure field, Mach number field and temperature field of compressible flow around the optimised blade geometry are shown, respectively. Again, Fig 24 shows that compressible flow through the turbine blade passage is in transonic speed range. In Fig 23 and Fig 25 a large drop in both pressure and temperature on the suction side of the blade can be observed, as well as one small localised temperature and pressure drop on the pressure side of the leading edge. Due to the Bernoulli principle, an increase in velocity can be observed, both on the suction side and localised on the pressure side of the leading edge. In Fig 26 turbulent kinetic energy field with the detail of the trailing edge can be observed. A wake behind the optimised blade can be observed both in Fig 26, and in Fig 27. Due to the fact that value of an attack angle was 2.5^o and the wake behind this blade is directed downwards, a conclusion that positive lift is generated can be derived. Also in comparison with Fig 19, and Fig 20 lower maximal values of both turbulent kinetic energy and turbulent viscosity were obtained.

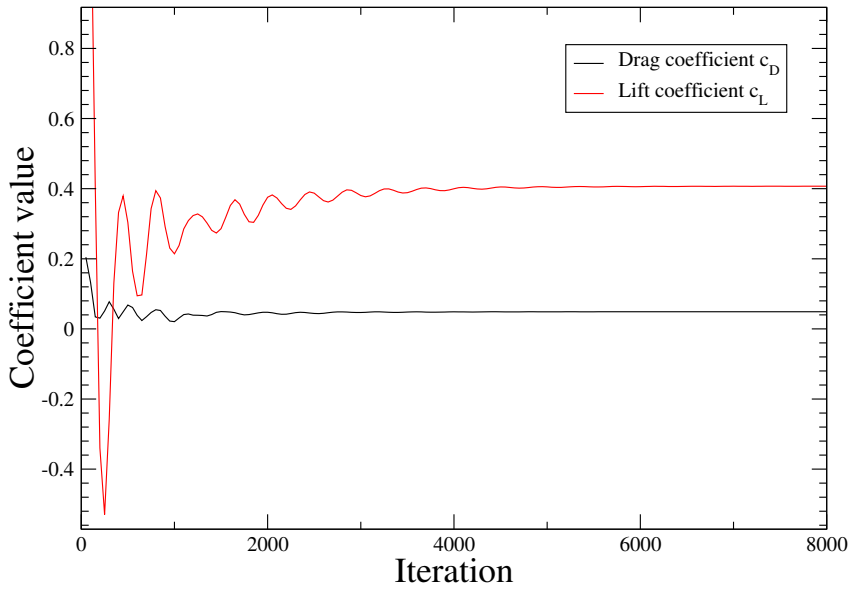


Figure 28: Convergence of the lift and the drag coefficient values for the attack angle $\alpha = 2.5^o$.

Convergence of the drag and lift coefficient can be observed in Fig 28. Calculated values of objective functions for optimised blade geometry at 2.5^o attack angle are as follows:

(a) $\varepsilon = 8.31016945$

(b) $c_D = 0.04896487$

During optimisation process 81 iterations were run and 12 generations within MOGA were performed.

6.7.2 Attack Angle $\alpha = 7.5^o$

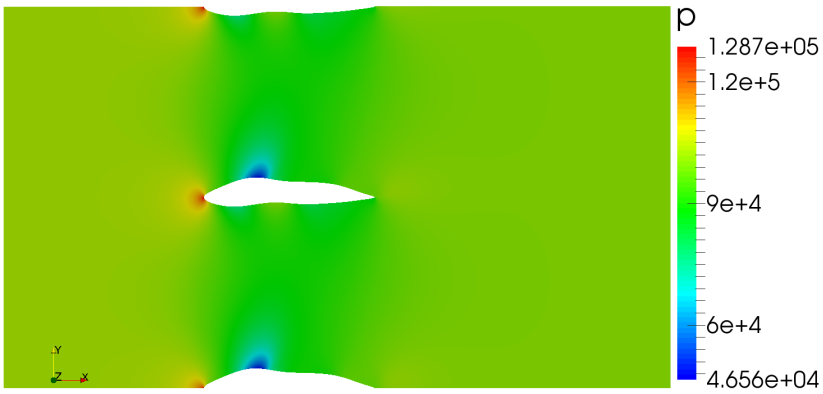


Figure 29: Pressure field for the initial blade geometry for the attack angle $\alpha = 7.5^o$.

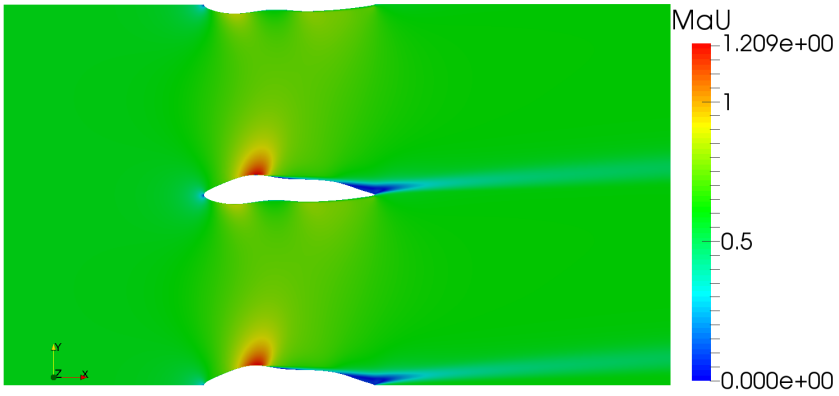


Figure 30: Mach number field for the initial blade geometry for the attack angle $\alpha = 7.5^o$.

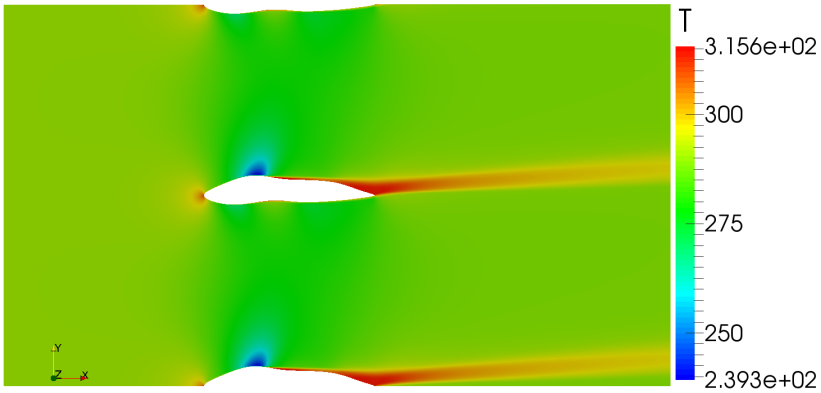


Figure 31: Temperature field for the initial blade geometry for the attack angle $\alpha = 7.5^o$.

Compressible flow was simulated for the initial blade geometry at 7.5^o attack angle fluid flow and pressure field, Mach number field, and temperature field were obtained as presented in Fig 29, Fig 30, and Fig 31, respectively. Due to the non-aerodynamic shape of the initial geometry, localised pressure and temperature drops along the pressure and suction side of the blade can be observed, as well as localised increases in fluid velocity. Localised pressure drops on the pressure side of the blade are considerably lower than 2.5^o attack angle, thus positive lift can be expected, but all stated before contributes to the conclusion that very low value of the lift force is to be expected. Again, Fig 30 shows that the fluid flow was in the transonic speed range.

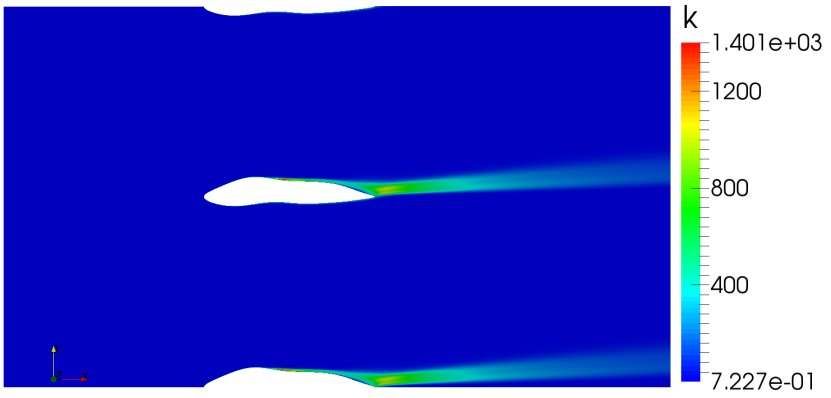


Figure 32: Turbulent kinetic energy field for the initial blade geometry for the attack angle $\alpha = 7.5^o$.

In Fig 32 turbulent kinetic energy field for fluid flow at 7.5^o attack angle can be seen and in Fig 33 the turbulent viscosity field. In both figures, wake behind the blade can be observed. In comparison with 2.5^o attack angle case, the wake is directed upwards under smaller angle, while at the same time attack angle is greater. This contributes to the fact that the generated lift force will be positive. Again, relatively high maximum values of both kinetic turbulent energy and turbulent viscosity were obtained.

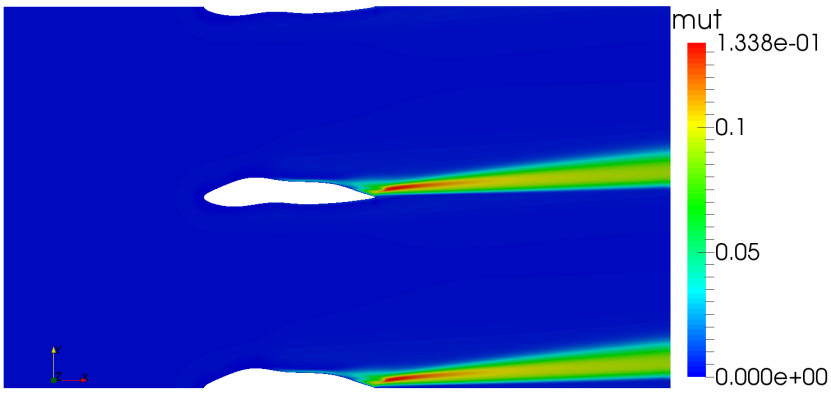


Figure 33: Turbulent viscosity field for the initial blade geometry for the attack angle $\alpha = 7.5^o$.

Calculated results of the objective functions are as follows:

(a) $\varepsilon = 2.564$

(b) $c_D = 0.0754$

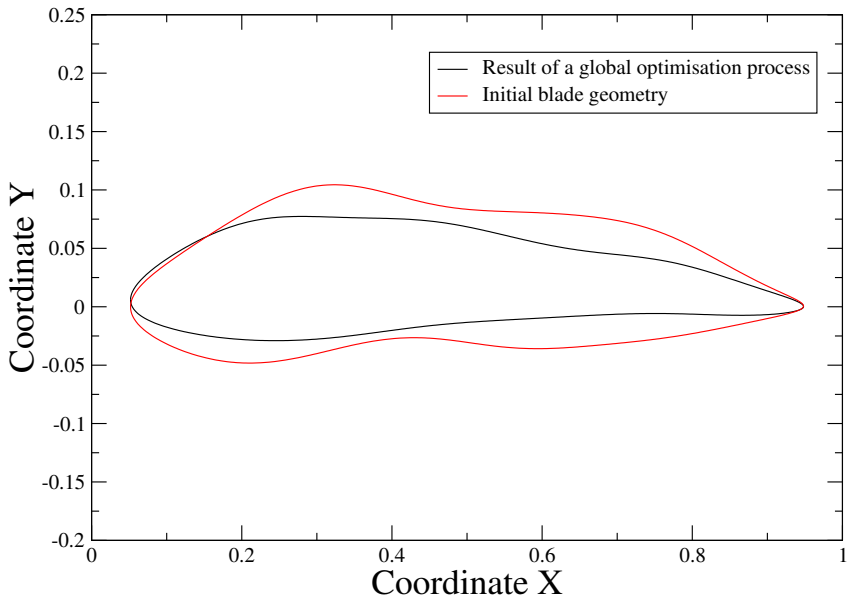


Figure 34: Comparison of the initial and the optimised blade geometry for attack angle $\alpha = 7.5^o$.

In Fig 34 comparison of the initial and optimised blade geometry is shown. This new blade geometry is a product of a global optimisation process. In Fig 35 position of control polygon vertices that define new, optimised, blade geometry within it's geometric bounds can be seen. Again, no discontinuities along the blade geometry can be found due to the favourable B-spline curve properties.

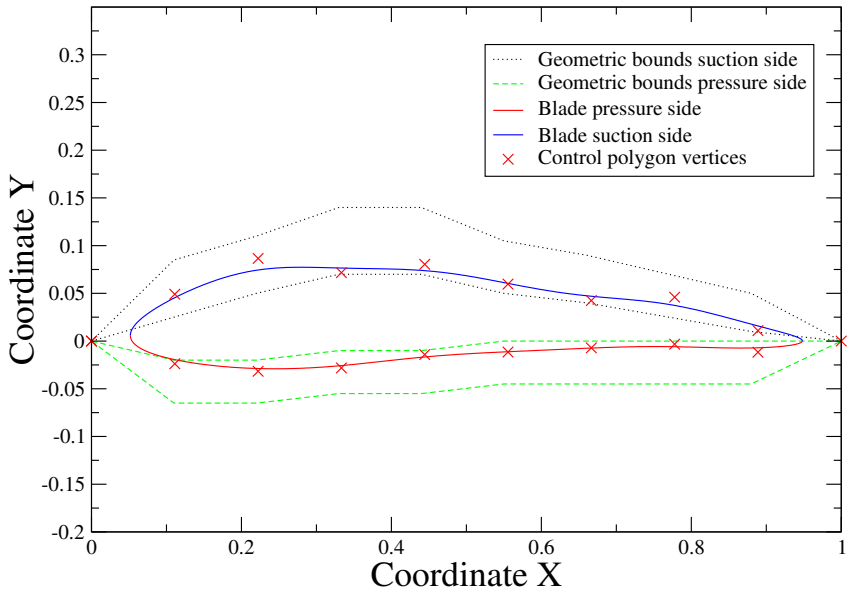


Figure 35: Optimised blade geometry for attack angle $\alpha = 7.5^o$.

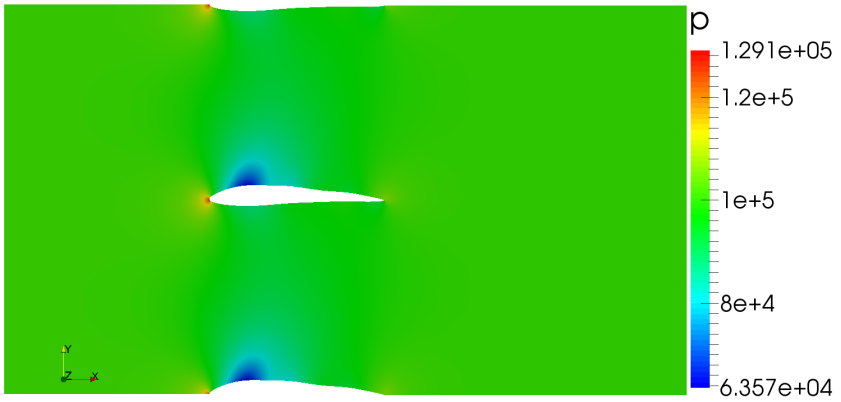


Figure 36: Pressure field for the optimised rotor blade geometry for the attack angle $\alpha = 7.5^o$.

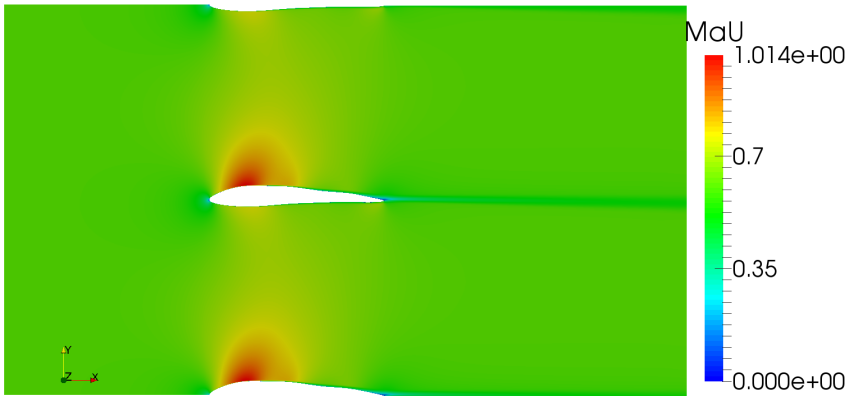


Figure 37: Mach number field for the optimised rotor blade geometry for the attack angle $\alpha = 7.5^o$.

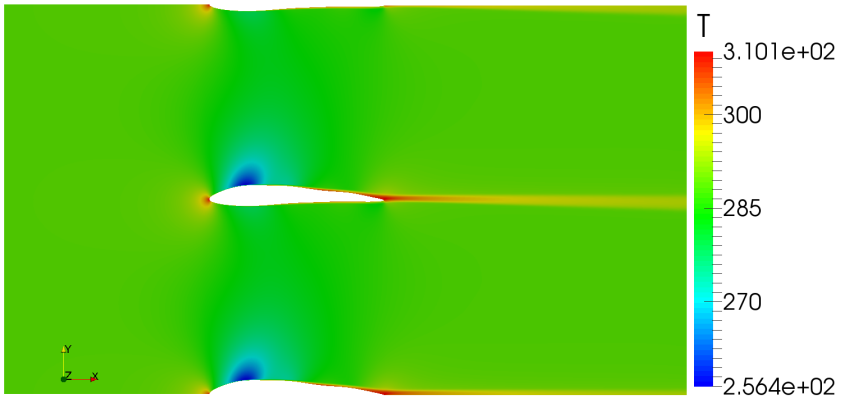


Figure 38: Temperature field for the optimised rotor blade geometry for the attack angle $\alpha = 7.5^o$.

In Fig 36, Fig 37, and Fig 38 pressure field, Mach number field, and temperature field are

shown, respectively. In all three cases it can be seen that this shape is quite favourable. Due to geometric bounding, as well as to boundary condition values, NACA-like turbine blade was expected to be a result of this stationary turbine blade passage optimisation. Fig 36 show that the velocity in streamwise direction along the suction side of the blade is significantly higher than what is the case in the rest of the computational domain. Thus, expected pressure and temperature drop can be observed in Fig 36, and Fig 38, respectively. Due to concentration of lower pressure on the suction side, respectable lift force is to be expected.

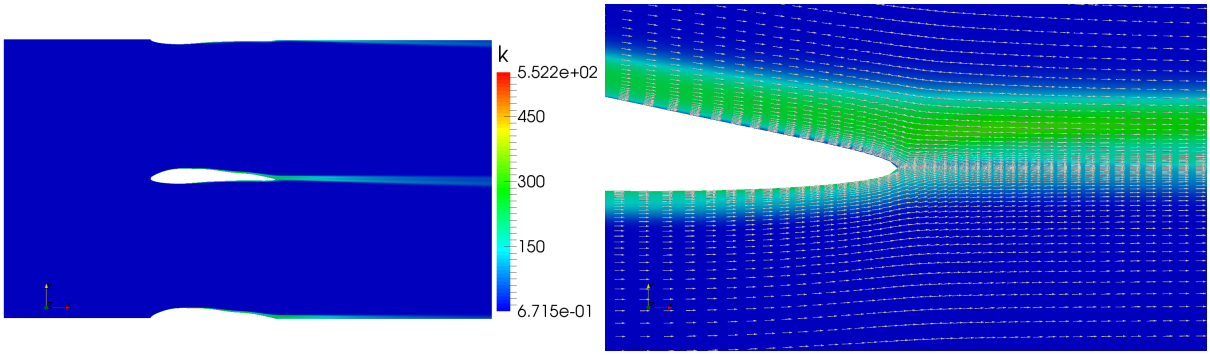


Figure 39: Turbulent kinetic energy field with detailed trailing edge for the optimised rotor blade geometry for the attack angle $\alpha = 7.5^o$.

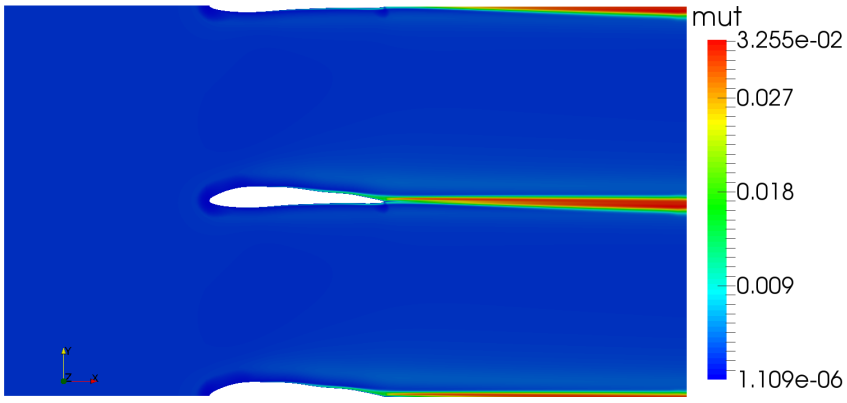


Figure 40: Turbulent viscosity field for the optimised rotor blade geometry for the attack angle $\alpha = 7.5^o$.

Fig 39 show turbulent kinetic energy field with detailed trailing edge, and the wake behind the optimised blade geometry is shown both in Fig 39, and Fig 40. Maximum values of both fields are significantly lower than those shown for the initial blade geometry. Also, wake behind the optimised blade is pointed downwards while the attack angle is 7.5^o. That indicates that this blade geometry is capable of producing a large lift force. Also, the wake behind this blade dissipates under relatively small angle, and the reason for that is, even though this is a transonic case, airflow separation is not pushed significantly towards the leading edge of the blade.

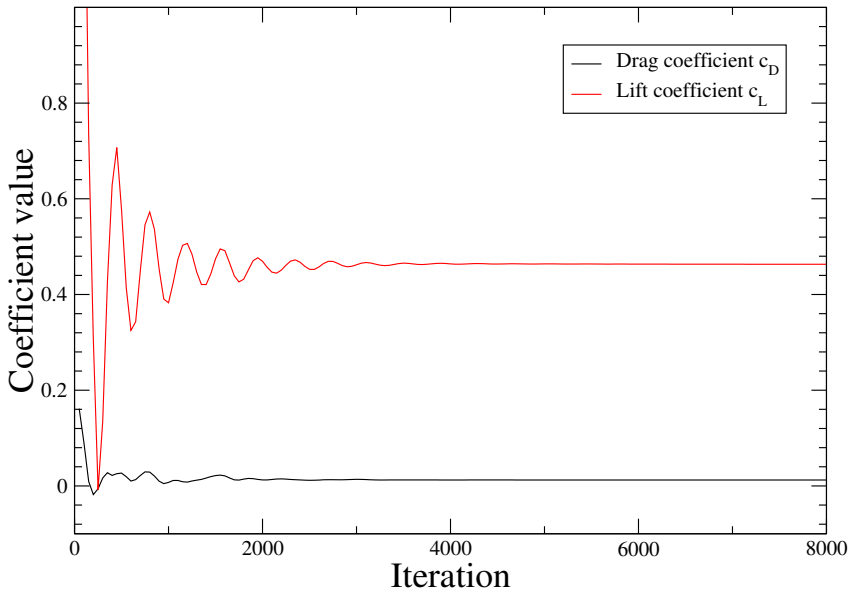


Figure 41: Convergence of the lift and the drag coefficient for optimised blade geometry for the attack angle $\alpha = 7.5^o$.

Fig 41 show convergence of lift and the drag coefficient values. In this particular case, 77 iterations of automated optimisation process was needed, and MOGA, produced 12 generations, before meting stopping criteria. Values of objective functions of this optimised blade geometry are as follows:

(a) $\varepsilon = 37.264$

(b) $c_D = 0.01243$

6.8 Results of Stator Blade Optimisation

In turbines, stator blades are vital parts of the geometry. They are used to accelerate the fluid flow as much as possible with smallest possible pressure drop, while directing fluid flow at optimal attack angle onto rotors. Having this in mind, objective functions were defined as follows. The target was to produce a geometry that would give the biggest increase in velocity while keeping pressure drop as small as possible. Thus Δu, and Δp were defined as:

$$\Delta u = u_{outlet} - u_{inlet}, \tag{53}$$

and

$$\Delta p = p_{inlet} - p_{outlet}. \tag{54}$$

First, calculated values for initial blade geometry will be given, and then for the optimised blade geometry.

6.8.1 Attack Angle $\alpha = 0^o$

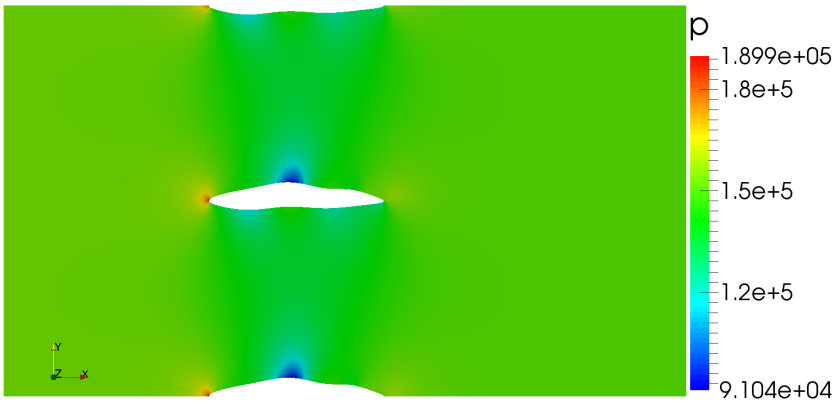


Figure 42: Pressure field for the initial blade geometry for the attack angle $\alpha = 0^o$.

In Fig 42, Fig 43, and Fig 44 pressure field, Mach number field and temperature field are presented, respectively. Shape of the initial geometry is the same as is was in the case of the rotor blade and it's non-aerodynamic shape produces inconherent fields. Localised pressure and temperature drops on both pressure and the suction side of the blade, and therefore sudden acceleration of the fluid flow introduces additional disturbances.

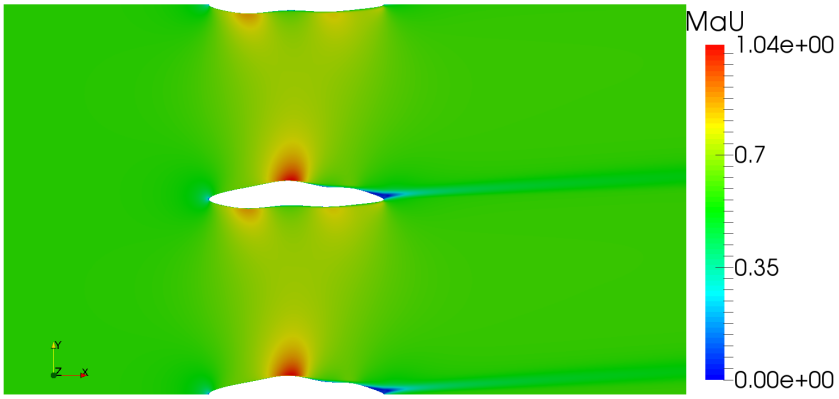


Figure 43: Mach number field for the initial blade geometry for the attack angle $\alpha = 0^o$.

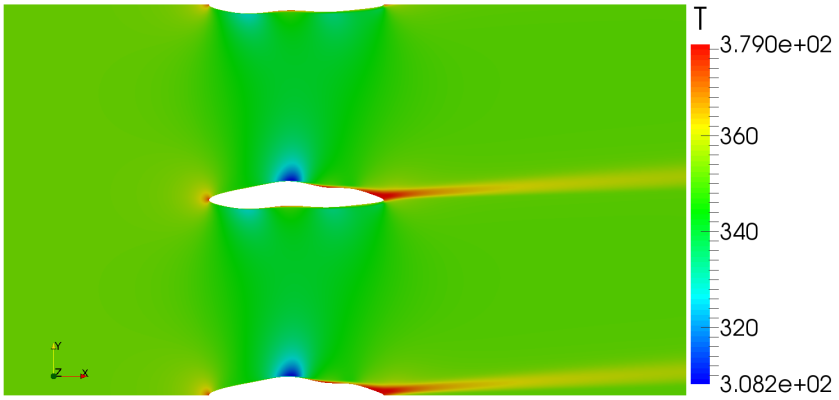


Figure 44: Temperature field for the initial blade geometry for the attack angle $\alpha = 0^o$.

Fig 43, and Fig 44 show that this geometry directs fluid flow upwards.

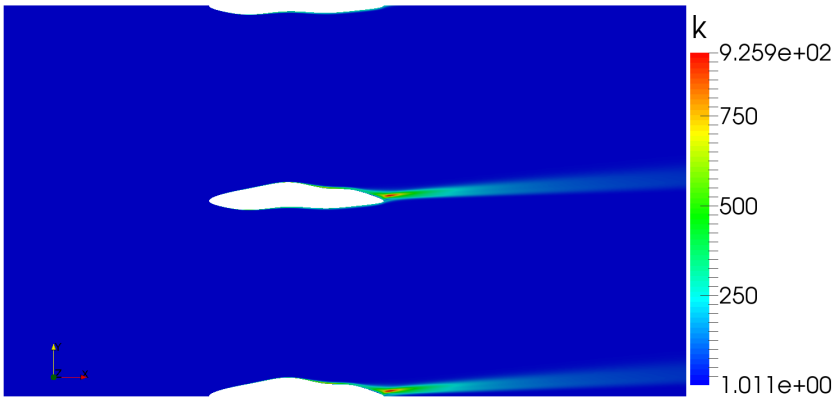


Figure 45: Turbulent kinetic energy field for the initial blade geometry for the attack angle $\alpha = 0^o$.

In Fig 45 which shows turbulent kinetic energy field, and Fig 46 which shows the turbulent viscosity field, it can be seen that the wake behind the blade geometry is relatively strong. In optimisation process of stator blade geometry, as stated before, a maximal increase of velocity was the objective while keeping the pressure drop as small as possible. Results obtained for this blade geometry are as follows:

(a) $\Delta p = 2493.79$ Pa

(b) $\Delta U = 3.561$ m/s

(c) $\beta = 3.47^o$

where β is the angle at which the fluid flow passes through the outlet in respect to the x axis.

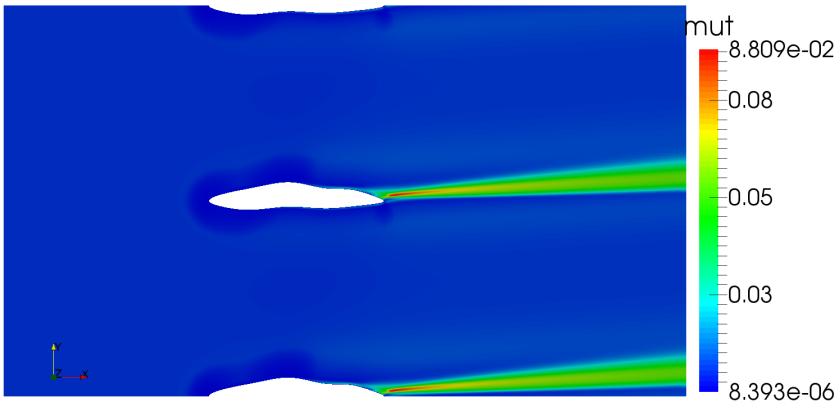


Figure 46: Turbulent viscosity field for the initial blade geometry for the attack angle $\alpha = 0^o$.

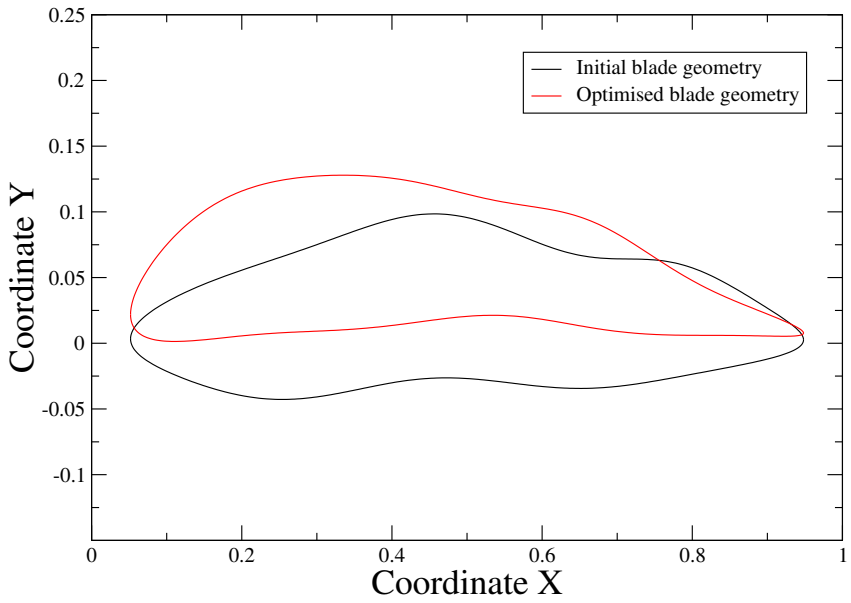


Figure 47: Comparison of the initial blade geometry and the optimised blade geometry for the attack angle $\alpha = 0^o$.

In Fig 47 comparison of the initial and the optimised blade geometry can be observed, while Fig 48 shows detailed optimised blade geometry with control polygon vertices within geometric bounds. In the optimisation process for the 0^o attack angle, 517 iterations were performed before stopping criteria of 10 (maximal number) generations was met.

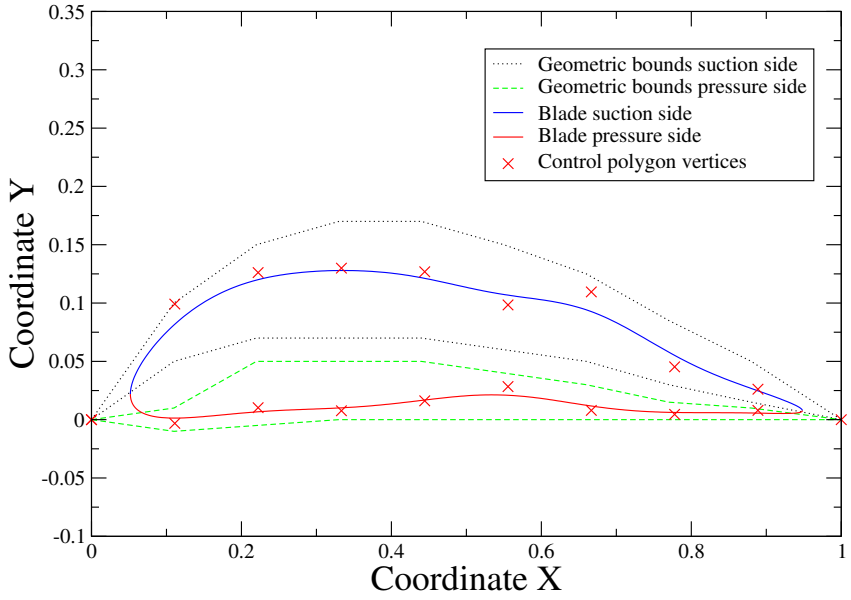


Figure 48: Optimised stator blade geometry for the attack angle $\alpha = 0^o$.

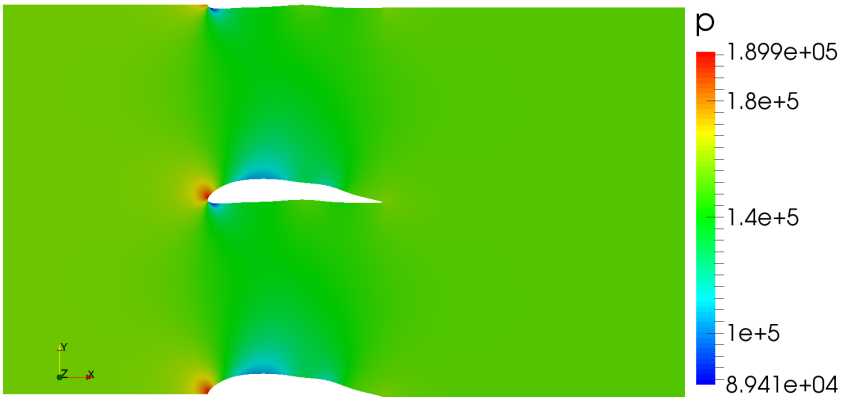


Figure 49: Pressure field for the optimised stator blade geometry for the attack angle $\alpha = 0^o$.

In Fig 49, Fig 50, and Fig 51 the pressure field, Mach number field and the temperature field can be observed respectively. On the suction side of the blade there is significant area of lowered pressure that induces increase in the fluid velocity, and therefore better stator blade performance. It can be seen that the fluid exiting the blade is directed downwards. The pressure drop on the leading edge of the optimised stator blade is a result of the attack angle because above this area separation of the fluid flow happens.

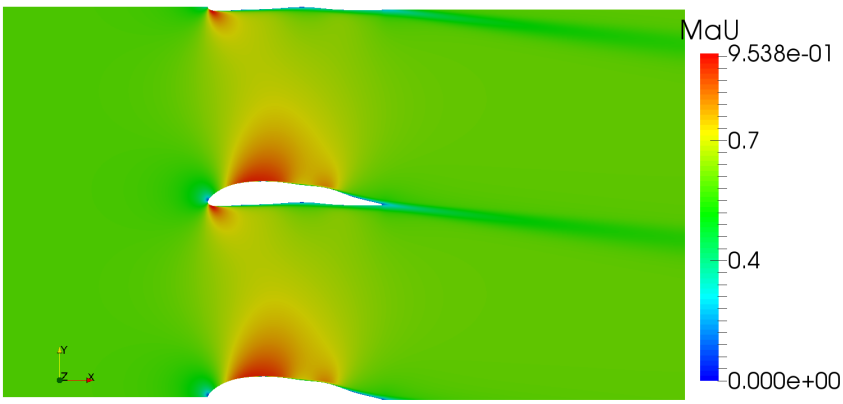


Figure 50: Mach number field for the optimised stator blade geometry for the attack angle $\alpha = 0^o$.

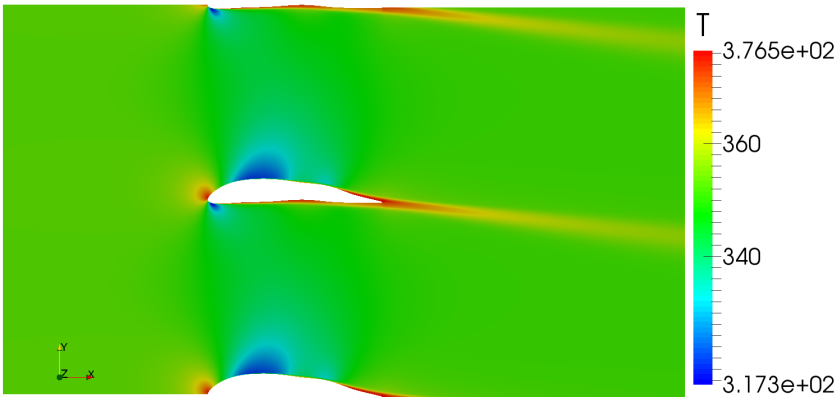


Figure 51: Temperature field for the optimised stator blade geometry for the attack angle $\alpha = 0^o$.

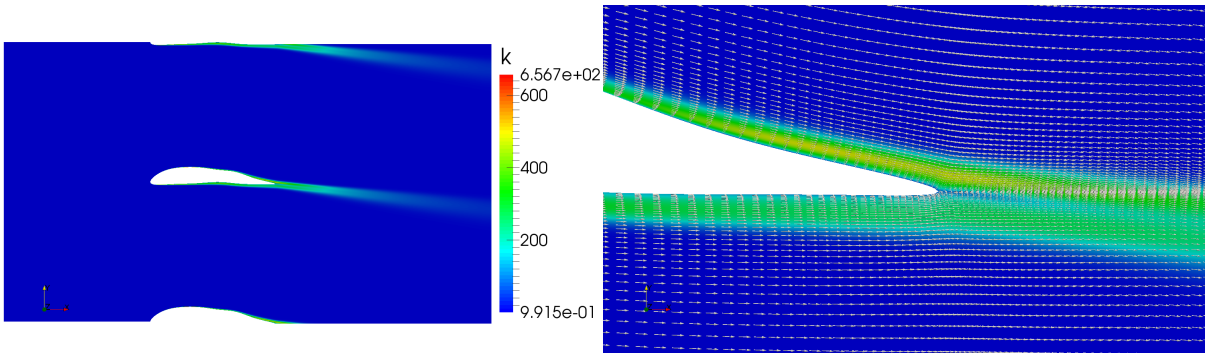


Figure 52: Turbulent kinetic energy field with detailed trailing edge for the optimised stator blade geometry for the attack angle $\alpha = 0^o$.

In Fig 52 turbulent kinetic energy field with detail of the trailing edge is presented, while Fig 53 shows turbulent viscosity field. It can be seen that the maximum values of both parameters are lower in comparison with the initial blade geometry. Also, from the direction of the wake it can be deduced that the fluid flow is directed downwards. Fig 52 shows that the fluid flow in streamwise direction leaves the trailing edge without creating any recirculation zone on the suction side of the blade.

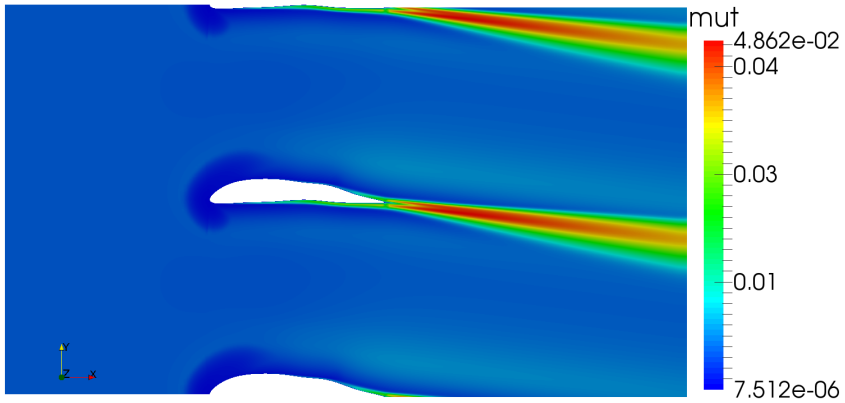


Figure 53: Turbulent viscosity field for the optimised stator blade geometry for the attack angle $\alpha = 0^o$.

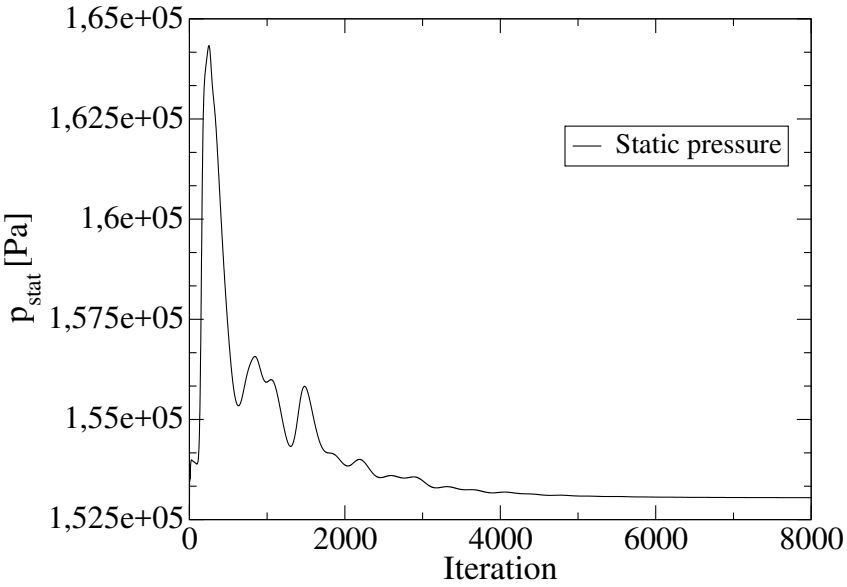


Figure 54: Convergence of static pressure value at the inlet of the domain for the attack angle $\alpha = 0^o$.

Figures Fig 54, and Fig 55 show convergence of the velocity and pressure values at the inlet and at the outlet of the numerical domain, respectively. Calculated values of objective functions and the value of the angle at which fluid flow leaves the domain are as follows:

(a) $\Delta p = 3047.95$ Pa

(b) $\Delta U = 5.253$ m/s

(c) $\beta = -6.909^o$

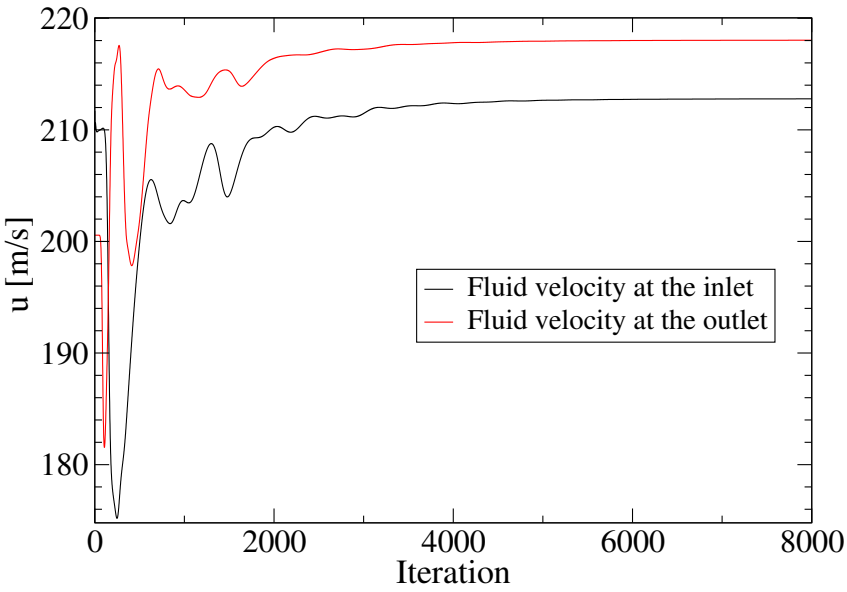


Figure 55: Convergence of velocity values at the inlet and at the outlet of the domain for the attack angle $\alpha = 0^o$.

6.8.2 Attack Angle $\alpha = 5^o$

First, calculated results for the initial blade geometry will be shown, and then the results for the optimised stator blade geometry.

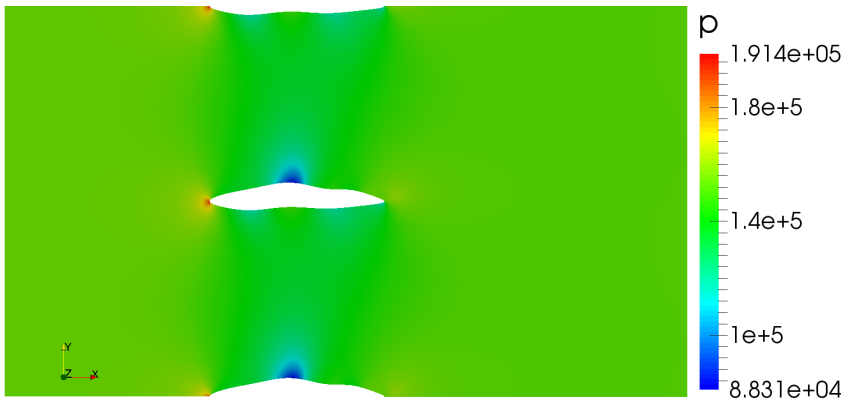


Figure 56: Pressure field for the initial blade geometry for the attack angle $\alpha = 5^o$.

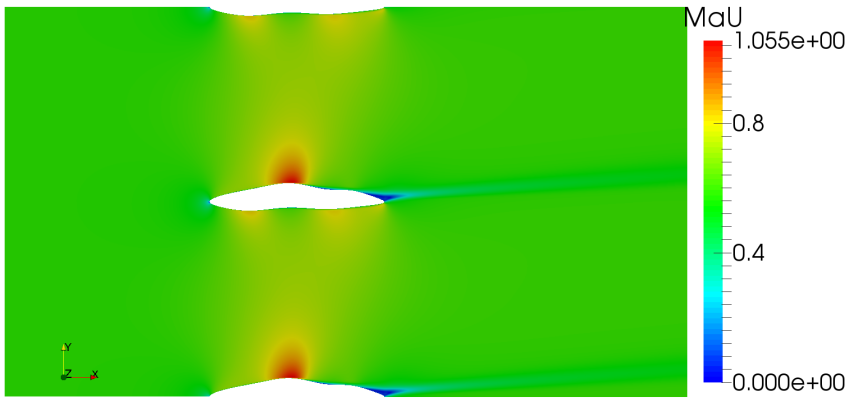


Figure 57: Mach number field for the initial blade geometry for the attack angle $\alpha = 5^o$.

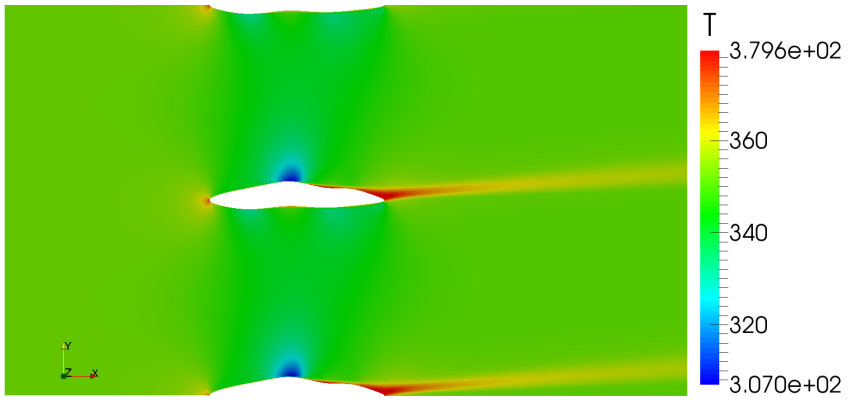


Figure 58: Temperature field for the initial blade geometry for the attack angle $\alpha = 5^o$.

Fig 56, Fig 57, and Fig 58 show the pressure field, Mach number field, and temperature field for initial blade geometry, respectively. From Fig 57 it can be seen that the fluid flow is in transonic speed range, and Fig 57, and Fig 58 show that the fluid flow leaving the blade is directed upwards. Pressure and temperature drops exist on both pressure and the suction side of the blade, and therefore localised velocity increase in the same location can be expected. Maximal value of the turbulent kinetic energy is quite high and the peak in the value is concentrated close to the trailing edge, shown in Fig 59.

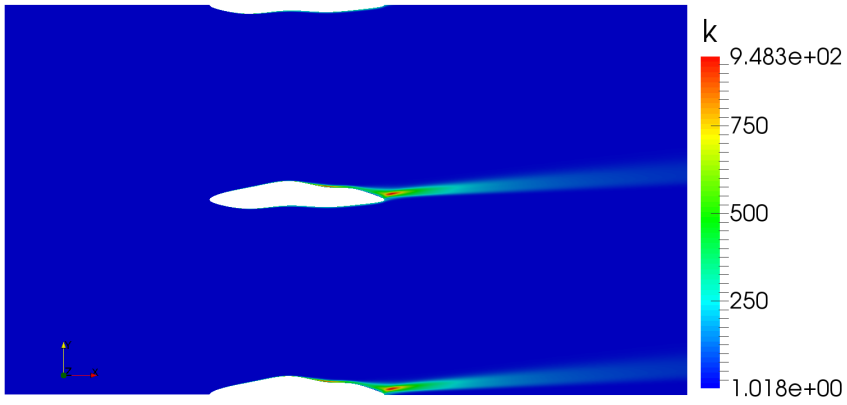


Figure 59: Turbulent kinetic energy field for the initial blade geometry for the attack angle $\alpha = 5^o$.

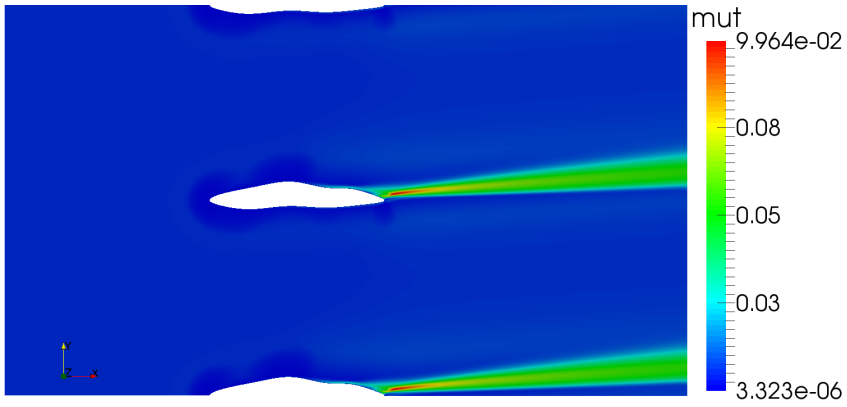


Figure 60: Turbulent viscosity field for the initial blade geometry for the attack angle $\alpha = 5^o$.

Calculated values of the objective functions for the initial blade geometry at the 5^o attack angle are as follows:

(a) $\Delta p = 2251.37$ Pa

(b) $\Delta U = 2.695$ m/s

(c) $\beta = 4.132^o$

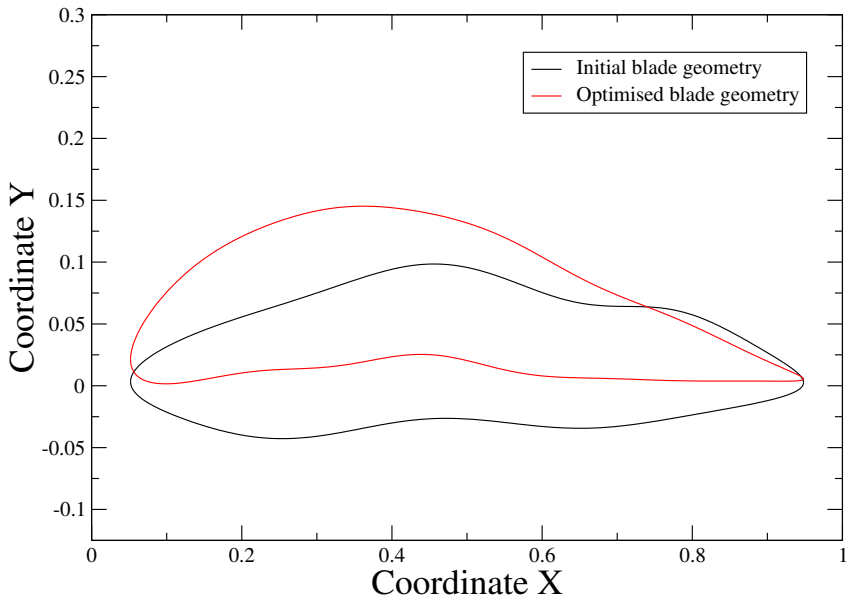


Figure 61: Comparison of the initial and the optimised stator blade geometry for the attack angle $\alpha = 5^o$.

In Fig 61 comparison of the initial blade geometry and optimised stator blade geometry for 5^o attack angle can be observed. The optimisation process ran 887 iterations before meeting the stopping criteria of 10 generation in MOGA. In Fig 62 optimised stator blade geometry for given geometric bounds can be seen.

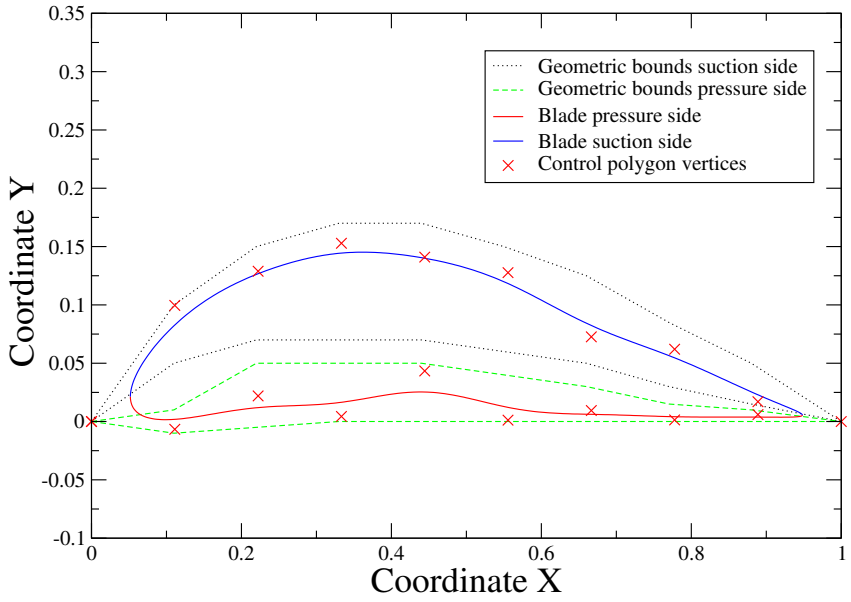


Figure 62: Optimised stator blade geometry for the attack angle $\alpha = 5^o$.

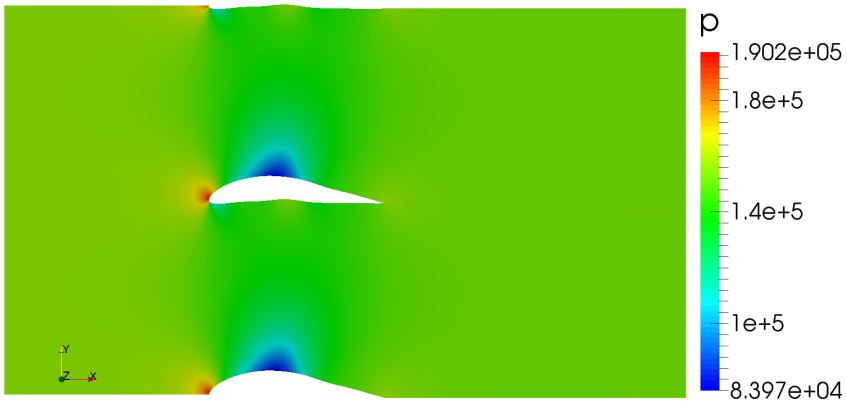


Figure 63: Pressure field for the optimised stator blade geometry for the attack angle $\alpha = 5^o$.

Fig 63, Fig 64, and Fig 65 show pressure field, Mach number field and temperature field, respectively. From Fig 64 it can be seen that this compressible fluid flow is in transonic speed range. Pressure and temperature drops are located on the suction side of the blade and they cover a significant blade area. On the same location, significant speed increase, due to the Bernoulli principle, in respect to the rest of the domain can be observed. Fluid flow is directed downwards, what is shown in Fig 66, and Fig 67 by direction of the wake behind the blade.

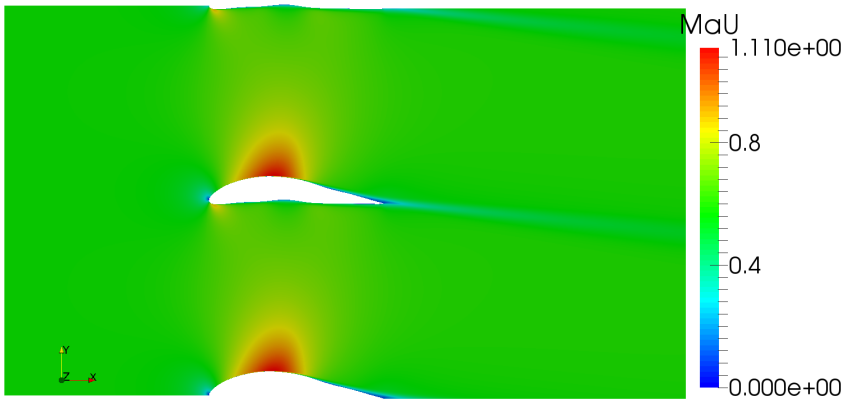


Figure 64: Mach number field for the optimised stator blade geometry for the attack angle $\alpha = 5^o$.

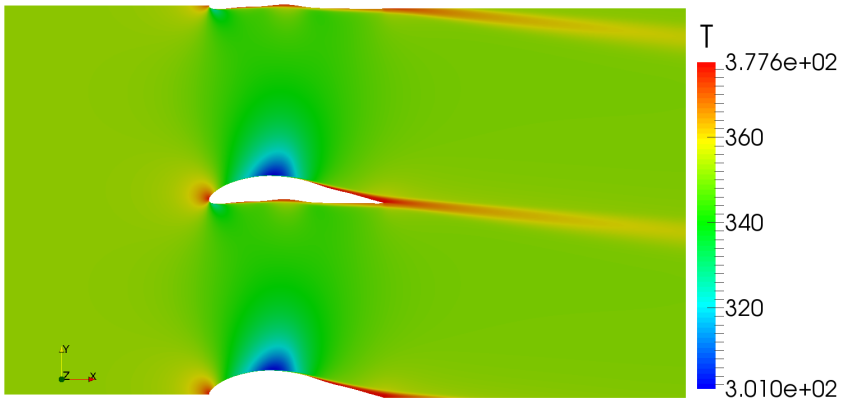


Figure 65: Temperature field for the optimised stator blade geometry for the attack angle $\alpha = 5^o$.

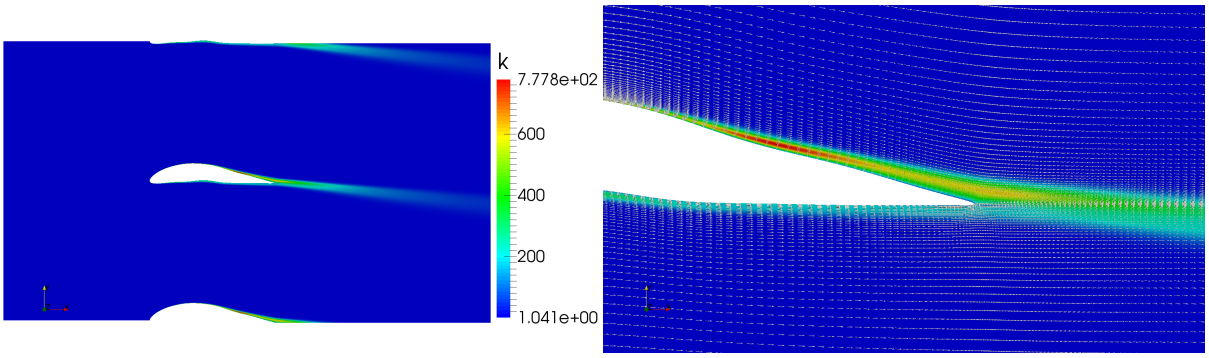


Figure 66: Turbulent kinetic energy field with detailed trailing edge for the optimised stator blade geometry for the attack angle $\alpha = 5^o$.

In Fig 66 turbulent kinetic energy field with detail of trailing edge can be seen while Fig 67 shows turbulent viscosity field. It can be seen that the maximal values of both fields are lower in comparison to the values calculated for the initial blade geometry.

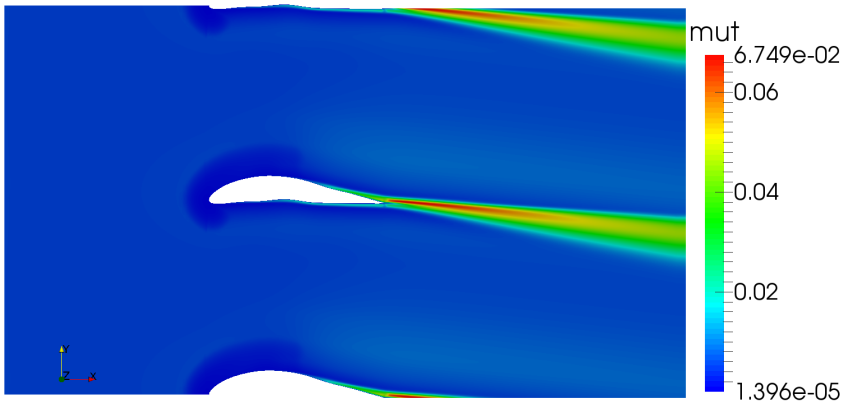


Figure 67: Turbulent viscosity field for the optimised stator blade geometry for the attack angle $\alpha = 5^o$.

In Fig 66 detailed turbulent kinetic energy field along the trailing edge of the optimised stator blade can be observed. Also, peak of the value is located close to the trailing edge but directed in counter-streamwise direction. Thus, this stator blade would create a recirculation zone from the pressure to the suction side of the blade if the attack angle would be increased.

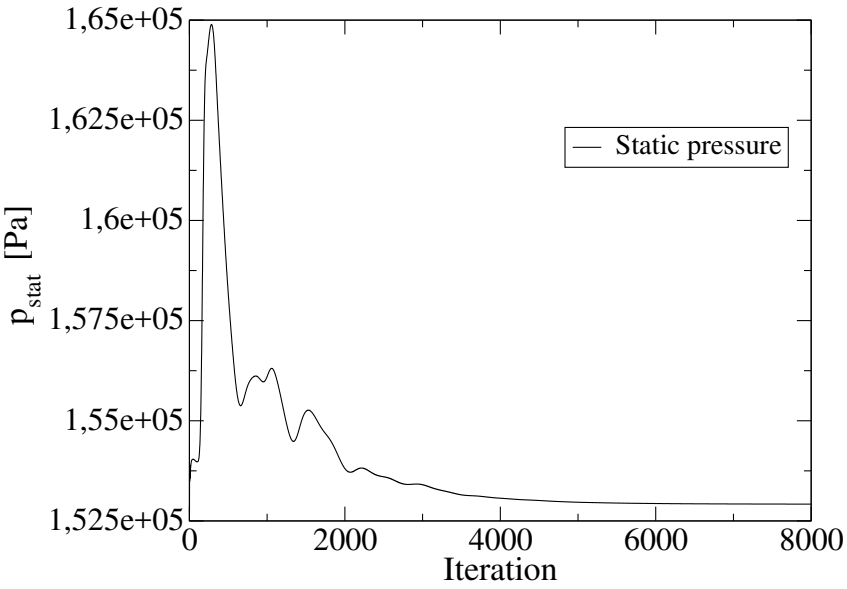


Figure 68: Convergence of the static pressure value at the inlet of a numerical domain for optimised stator blade geometry for the attack angle $\alpha = 5^o$.

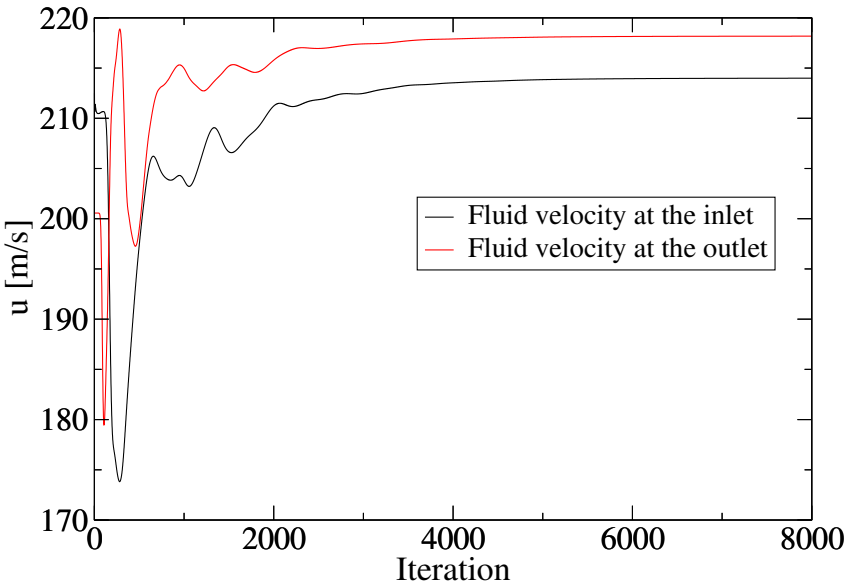


Figure 69: Convergence of velocity values at the inlet and at the outlet of a numerical domain for optimised stator blade geometry for the attack angle $\alpha = 5^o$.

Figures Fig 68, and Fig 69 show the convergence of the velocity and static pressure values at the inlet and at the outlet. Using Eq. (53), and Eq. (54) objective function values were calculated. Obtained values of objective functions and the angle at which fluid flow is exiting the domain are as follows:

(a) $\Delta p = 2919.64$ Pa

(b) $\Delta U = 4.185$ m/s

(c) $\beta = -6.377^o$

Pareto Front

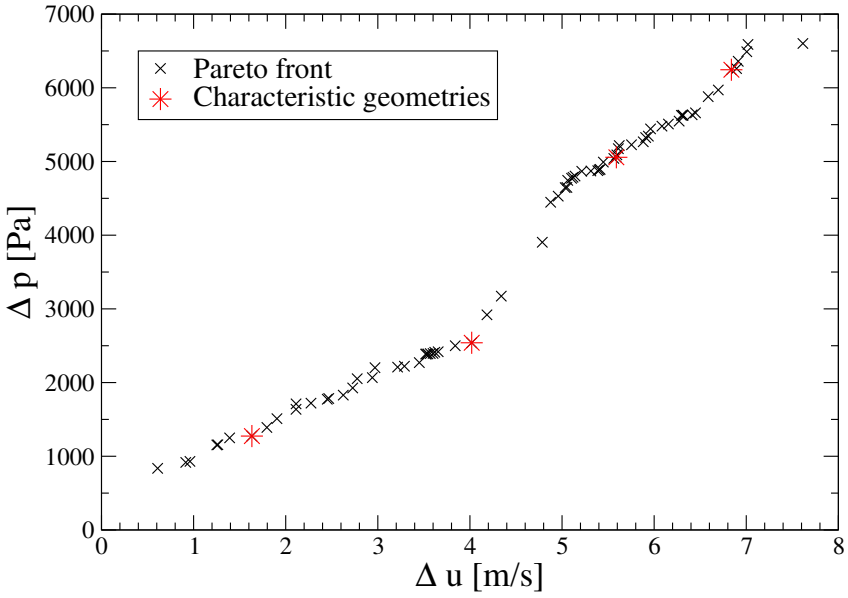


Figure 70: Pareto front for the stator blade geometry optimisation for the attack angle $\alpha = 5^o$.

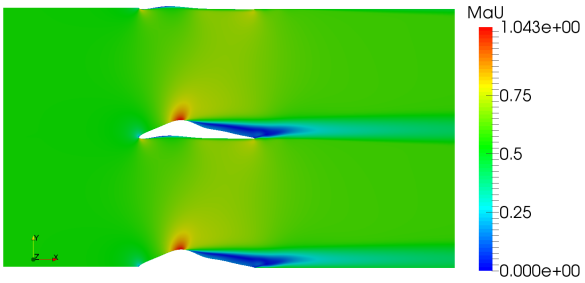


Figure 71: Iteration 160.

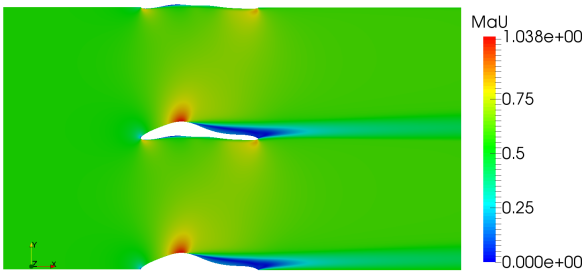


Figure 72: Iteration 887.

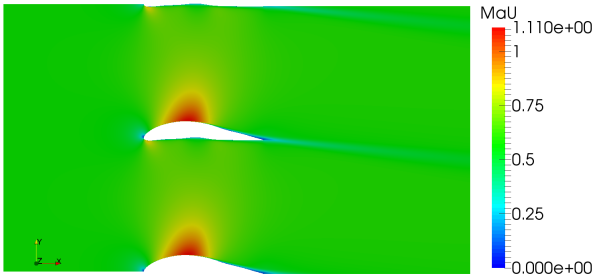


Figure 73: Iteration 6.

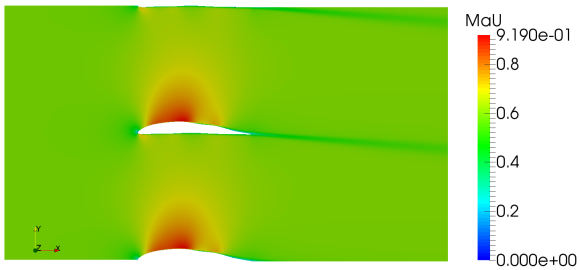


Figure 74: Iteration 590.

In Fig 70 a pareto front for the stator blade geometry optimisation for an attack angle $\alpha = 5^o$

is shown. Red stars denote position of four characteristic geometries along the pareto front and they are shown in Fig 71, Fig 72, Fig 73, and Fig 74, respectively. Objective function values of these four characteristic geometries are presented in Tab 8.

Table 8: Parameters of four characteristic Pareto front geometries.

Iteration of optimisation process	**Point on the pareto front**	**Objective function values**
160	68	$\Delta u = 6.839$ m/s $\Delta p = 6245.39$ Pa
887	49	$\Delta u = 5.589$ m/s $\Delta p = 5057.05$ Pa
6	30	$\Delta u = 4.185$ m/s $\Delta p = 2919.64$ Pa
590	6	$\Delta u = 1.633$ m/s $\Delta p = 1274.37$ Pa

Pareto front is obtained due to the nature of multi-objective optimisation algorithm. Optimisation process is searching for the optimal value of each objective function, and therefore number of solutions are proposed. As stated before, an optimal stator blade will be the one that can produce largest increase in velocity while keeping the pressure drop as small as possible. This is the reason why blade geometry produced in sixth iteration was chosen as an optimal one.

6.9 Closure

In this chapter, workflow of the developed automated optimisation process for turbine blade was presented, as well as the optimisation results. The workflow consists of four steps: geometry parametrisation, morphing (deformation) of the initial computational mesh, numerical calculation of compressible flow through the blade passage and creating a new set of geometry points by the optimisation genetic algorithm. B-spline was used for geometry parametrisation because it generates a smooth geometry, without any jumps or discontinuities. However, it is obvious from the results that parametrisation is the bottleneck of the workflow. The geometry is represented by a set of points which directly describe the shape of the blade, and these points are given by the optimisation algorithm as design variables. Strong geometric constraints must be prescribed to narrow the search area in the optimisation algorithm to make it fast and robust. More flexible, but equally efficient algorithm could be achieved if blade geometric parameters were parametrised, rather than the shape of the blade. For example, camber line could be parametrised and its curvature or maximum radius location could be the design variables. The current optimisation process did produce good global optimal solutions, but engineering experience was needed to obtain these solutions. The second bottleneck,

creating the computational mesh, was successfully resolved by implementing a mesh morphing algorithm. An initial mesh was used and deformed to suit the new geometry and applied in the numerical calculation of the flow through the blade passage. In this way, discretisation errors coming from mesh quality and mesh resolution were localised for all blade geometries. It is important to note the benefits of the mesh morphing in comparison to other options. Mesh morphing takes approximately 100 seconds per case, and mesh quality and resolution are controlled by using the same initial mesh for all cases. Using a hand generated mesh enables control over the mesh quality and resolution, but it takes 25 minutes to be finished. Automatic mesh generators can produce a mesh in 5 minutes, but there is no control over discretisation errors. A genetic algorithm was used for the optimisation and it needs a large number of estimates to converge to an optimal solution. That is why, as mentioned before, the search area was narrowed by employing strong geometric constraints. The constraints also prevented the solution to converge to a flat plate. The final results of the optimisation process for rotor and stator blades were presented. Global optimal solutions were obtained. It can be noticed that a blade similar to NACA family was obtained for 7.5^o attack angle flow on a rotor blade. For stator blade, the best result was produced for 5^o attack angle, but the geometry could be further improved by using local optimisation methods. For future work, it would be beneficial to include the attack angle as one of the design parameters and also to create a boundary condition which would fix the outlet velocity direction for the stator blade. But the biggest improvement would be the parametrisation of the blade parameters (camber angle, blade exit angle, camber length, etc.) which could then be used as design parameters.

7 Conclusion and Future Work

This thesis describes a development of automated optimisation process for turbine blade geometry. From the work presented here it can be concluded that this optimisation process can transform inconherent blade geometry into stator and rotor blade geometries. Therefore, expensive experiments can be replaced by computer aided optimisation process.

Optimisation process has a number of favourable properties incorporated. For geometry parametrisation, periodic B-spline curve was used which does not interpolate control polygon vertices, but rather approximates them. This way, discontinuities along the blade surface were avoided. To assure that blade geometries would be comparable one to another, all control polygon vertices were equidistant, and position of two control vertices defining the leading and trailing edge were fixed. Thus, blades had the same chord length. For all numerical calculations, same initial mesh was used. Local mesh resolution and mesh quality are the two principle errors of discretisation in CFD. By reusing the same initial mesh, and morphing it to the new geometry, results are more easily comparable, due to the fact that different magnitudes of discretisation induced errors are disabled. In the optimisation step it was ensured that initial population, or initial number of inspected blade geometrics, consists of the same initial solutions. This all contributed to the fact that this optimisation process proved to be robust.

For the stator blade and for the rotor blade optimisation, a sufficient number of iterations for full convergence of examined objective function values was assured in numerical calculation of compressible flow.

Results obtained within this thesis show that this optimisation process can be used for 2D cases to find an optimal global optimisation solution. Further investigation by implementing derivative free local optimisation method on the product of a global optimisation process could prove to be quite useful. On the other hand, this additional step can not be integrated in an automated process because one needs to investigate results of the Pareto front before deducing which result is to be considered optimal. For a stator blade optimisation, more accurate optimisation process could be obtained if outlet velocity boundary condition with definable exiting fluid flow angle is developed.

The most important conclusion for future work, derived from the obtained results is the fact that this process could be expanded to optimise 3D geometries. Vast application could be found in the department of naval architecture, aviation etc. The reasoning behind this conclusion is the fact that for these problems, both global optimisation method and a robust optimiser have to be used.

References

[1] Jasak, H., "Numerical Solution Algorithms for Compressible Flows Lecture Notes," 2007.

[2] Jasak, H., "Numeričke metode u mehanici kontinuuma (Numerical Methods in Continuum Mechanics), Material for lectures," 2006.

[3] David F. Rogers, "An Introduction to NURBS With Historical Perspective," 2001.

[4] Škurić, V. "Multi-Objective Optimisation of a Tube Bundle Heat Exchanger," 2014.

[5] Janiga, G. Thévenin "Optimization and Computational Fluid Dynamics," 2008.

[6] Brian M. Adams, Mohamed S. Ebeida, Michael S. Eldred, John D. Jakeman, Kathryn A. Maupin, Jason A. Monschke, Laura P. Swiler, J. Adam Stephens, Dena M. Vigil, Timothy M. Wildey, William J. Bohnhoff, Keith R. Dalbey, John P. Eddy, Russell W. Hooper, Kenneth T. Hu, Patricia D. Hough, Elliott M. Ridgway, Ahmad Rushdi "Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.4 User's Manual," 2016.

[7] Brian M. Adams, Mohamed S. Ebeida, Michael S. Eldred, John D. Jakeman, Kathryn A. Maupin, Jason A. Monschke, Laura P. Swiler, J. Adam Stephens, Dena M. Vigil, Timothy M. Wildey, William J. Bohnhoff, Keith R. Dalbey, John P. Eddy, Russell W. Hooper, Kenneth T. Hu, Patricia D. Hough, Elliott M. Ridgway, Ahmad Rushdi "Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.4 Reference Manual," 2016.

[8] "CFD Online," `https://www.cfd-online.com/`, last access 17.01.2017.

[9] Cox, M.G. "The numerical evaluation of B-splines,National Physical Laboratory DNAC 4" 1971.

[10] de Boor, C., "On calculation with B-splines, Jour. Approx. Theory, Vol.6, pp. 50-62," 1972.

[11] Schoenberg, I.J. "Contributions to the problem of approximation of equidistant data by analytic functions,Q. Appl. Math., Vol. 4, pp. 45-99; pp. 112-141" 1946.

[12] Cohen, E., Riesenfeld, R.F., "General matrix representation for Beźier and B-spline curves, Comp. in Indus., Vol.3, pp. 9-15," 1982.

[13] Barbarić, M; Guzović, Z. "Optimal design of hydrokinetic turbines for river application, Digital Proceedings of the 11th Conference on the Sustainable Development on Energy, Water and Environment Systems – SDEWES, Lisabon, Portugal." 2016.

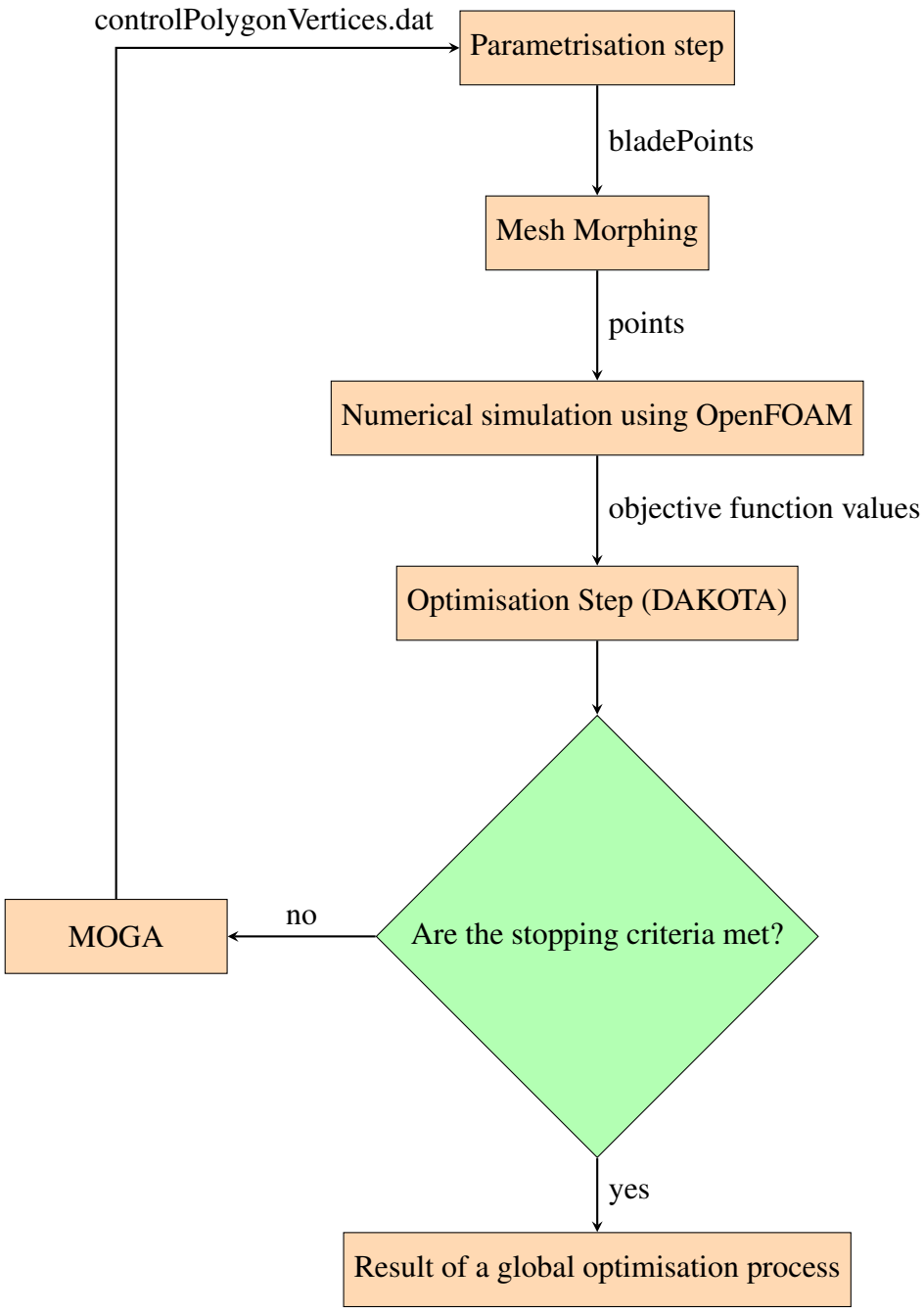
[14] Jasak, H. "Error analysis and estimation for the Finite Volume Method with applications to fluid flows., PhD Thesis, Imperial College of Science, Technology and Medicine, London," 1996.

[15] Tuković, Ž. "Metoda kontrolnih volumena na domenama promjenjivog oblika, PhD Thesis, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb," 2005.

[16] Beaudoin, M., Jasak, H. "Development of a Generalized Grid Interface for Turbomachinery simulations with OpenFOAM, Open Source CFD International Conference, Berlin, Germany" 2008.

Appendix A

Automated optimisation process workflow:



bladePoints

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.4                                   |
|   \\  /    A nd           | Web:      http://www.openfoam.org               |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

FoamFile
{
    version         2.0;
    format          ascii;
    class           dictionary;
    object          bladePoints;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

pressureSidePoints
(
(x1 y1 z1)
(x2 y2 z2)
.
.
.
(xn yn zn));

suctionSidePoints
(
(x1 y1 z1)
(x2 y2 z2)
.
.
.
(xn yn zn));
```

points

```
/*---------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | foam-extend: Open Source CFD                    |
|  \\    /   O peration     | Version:     4.0                                |
|   \\  /    A nd           | Web:         http://www.foam-extend.org         |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       vectorField;
    location    "2/polyMesh";
    object      points;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //


n
(
(x1 y1 z1)
(x2 y2 z2)
.
.
.
(xn yn zn)
)


// ************************************************************************* //
```

controlPolygonVertices.dat

```
0 0 0
0.1111111111 -y1 0
0.2222222222 -y2 0
0.3333333333 -y3 0
0.4444444444 -y4 0
0.5555555556 -y5 0
0.6666666667 -y6 0
0.7777777778 -y7 0
0.8888888889 -y8 0
1 0 0
0.8888888889 y9 0
0.7777777778 y10 0
0.6666666667 y11 0
0.5555555556 y12 0
0.4444444444 y13 0
0.3333333333 y14 0
0.2222222222 y15 0
0.1111111111 y16 0
0 0 0
```

Appendix B

dakota.in file for rotor blade optimisation

```
# Dakota Input File: dakota.in

strategy
    single
    tabular_graphics_data

method
moga
    id_method = 'MOGA'
    seed = 12345
    initialization_type unique_random
    population_size = 50
    crossover_type shuffle_random
      num_offspring = 40 num_parents = 50
      crossover_rate = 0.75
    mutation_type offset_normal
      mutation_rate = 1
      mutation_scale = 0.1
    fitness_type domination_count
    replacement_type below_limit = 2
      shrinkage_percentage = 0.25
    niching_type max_designs = 0 0
      num_designs = 150
    max_iterations = 30                  #Max. generations
    max_function_evaluations = 10000
    convergence_type
      metric_tracker
      percent_change 0.01
      num_generations 10

model
  single

variables
```

```
  continuous_design = 16
    lower_bounds     0.0100    0.0250    0.0400    0.0500
                     0.0700    0.0700    0.0500    0.0250
                     -0.065    -0.065    -0.055    -0.055
                     -0.045    -0.045    -0.045    -0.045
    initial_point    0.0250    0.0800    0.0050    0.0090
                     0.0110    0.0800    0.0600    0.0400
                     -0.025    -0.050    -0.040    -0.020
                     -0.030    -0.040    -0.025    -0.015
    upper_bounds     0.0500    0.0700    0.0900    0.1050
                     0.1400    0.1400    0.1100    0.0850
                     -0.020    -0.020    -0.010    -0.010
                     0.0000    0.0000    0.0000    0.0000
    descriptors      'y1'      'y2'      'y3'      'y4'
                     'y5'      'y6'      'y7'      'y8'
                     'y9'      'y10'     'y11'     'y12'
                     'y13'     'y14'     'y15'     'y16'

interface,
  fork
     analysis_driver = 'runAnalysis'
     asynchronous evaluation_concurrency = 2
  parameters_file = 'params.in'
  results_file = 'results.out'
        work_directory
          named 'Iteracija'
          directory_tag
          directory_save
  file_save

responses,
       objective_functions = 2
       descriptors = 'Cl_over_Cd' 'Cd'
       sense = "max" "min"
       no_gradients
       no_hessian
```

Appendix C

dakota.in file for stator blade optimisation

```
# Dakota Input File: dakota.in

strategy
    single
    tabular_graphics_data

method
moga
    id_method = 'MOGA'
    seed = 12345
    initialization_type unique_random
    population_size = 50
    crossover_type shuffle_random
      num_offspring = 40 num_parents = 50
      crossover_rate = 0.75
    mutation_type offset_normal
      mutation_rate = 1
      mutation_scale = 0.1
    fitness_type domination_count
    replacement_type below_limit = 1
      shrinkage_percentage = 0.25
    niching_type max_designs = 0 0
      num_designs = 150
    max_iterations = 10                  #Max. generations
    max_function_evaluations = 10000
    convergence_type
      metric_tracker
      percent_change 0.01
      num_generations 5

model
  single

variables
```

```
  continuous_design = 16
    lower_bounds     0.0150    0.0300    0.0500    0.0600
                     0.0700    0.0700    0.0700    0.0500
                     -0.010    -0.005    0.0000    0.0000
                     0.0000    0.0000    0.0000    0.0000
    initial_point    0.0250    0.0800    0.0050    0.0090
                     0.0110    0.0800    0.0600    0.0400
                     -0.025    -0.050    -0.040    -0.020
                     -0.030    -0.040    -0.025    -0.015
    upper_bounds     0.0500    0.0850    0.1250    0.1500
                     0.1700    0.1700    0.1500    0.1000
                     0.0100    0.0500    0.0500    0.0500
                     0.0400    0.0300    0.0150    0.0100
    descriptors      'y1'      'y2'      'y3'      'y4'
                     'y5'      'y6'      'y7'      'y8'
                     'y9'      'y10'     'y11'     'y12'
                     'y13'     'y14'     'y15'     'y16'

interface,
  fork
     analysis_driver = 'runAnalysis'
     asynchronous evaluation_concurrency = 4
  parameters_file = 'params.in'
  results_file = 'results.out'
        work_directory
          named 'Iteracija'
          directory_tag
          directory_save
  file_save

responses,
       objective_functions = 2
       descriptors = 'deltaU' 'deltaP'
       sense = "max" "min"
       no_gradients
       no_hessian
```